NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

MODELING TOOLKIT AND WORKBOOK FOR DEFENSE ANALYSIS STUDENTS

by

Chad A. Riden
Douglass M. Drake

September 2006

Thesis Advisor: Frank Giordano
Second Reader: Steve Horton

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 2006

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE Modeling Toolkit and Workbook for Defense Analysis Students

5. FUNDING NUMBERS

6. AUTHOR(S) Chad A. Riden, Douglass M. Drake

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

The purpose of this thesis is to provide a workbook to accompany the current textbook, *A First Course in Mathematical Modeling*. The workbook will offer basic reviews of each lesson followed by detailed examples of how to work each model. Topics covered include difference equations, systems of difference equations, Lanchester equations, graphical analysis, proportionality, geometric similarity, model fitting, Monte Carlo simulation, and probabilistic and deterministic behaviors. The thesis will also provide a modeling toolkit for the DA student upon graduation. The toolkit will present graduates simple instructions and multiple modeling templates they can take with them upon graduation and use to solve real-world modeling problems in the field. Templates in the toolkit cover decision theory, discrete dynamic systems, expected value, Lanchester models, and two-person games.

14. SUBJECT TERMS

Difference equations, Lanchester equations, graphical analysis, proportionality, geometric similarity, model fitting, Monte Carlo simulation, probabilistic, deterministic, decision theory, discrete dynamic systems, expected value, Lanchester models, two-person games.

15. NUMBER OF PAGES
133

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL

THIS PAGE INTENTIONALLY LEFT BLANK

MODELING TOOLKIT AND WORKBOOK FOR DEFENSE ANALYSIS STUDENTS

Chad A. Riden
Major, United States Air Force
B.S., Arizona State University, 1992

Douglass M. Drake
Major, United States Air Force
B.S., Jacksonville State University, 1988

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION OPERATIONS

from the

NAVAL POSTGRADUATE SCHOOL
September 2006

Authors: Chad A. Riden

Douglass M. Drake

Approved by: Frank Giordano
Thesis Advisor

Steve Horton
Second Reader

Gordon McCormick
Chairman, Department of Defense Analysis

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The purpose of this thesis is to provide a workbook to accompany the current textbook, *A First Course in Mathematical Modeling*. The workbook will offer basic reviews of each lesson followed by detailed examples of how to work each model. Topics covered include difference equations, systems of difference equations, Lanchester equations, graphical analysis, proportionality, geometric similarity, model fitting, Monte Carlo simulation, and probabilistic and deterministic behaviors. The thesis will also provide a modeling toolkit for the DA student upon graduation. The toolkit will present graduates simple instructions and multiple modeling templates they can take with them upon graduation and use to solve real-world modeling problems in the field. Templates in the toolkit cover decision theory, discrete dynamic systems, expected value, Lanchester models, and two-person games.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I need to first thank Dr. “General G” Giordano for offering this thesis to me. Further, without Dr. Giordano’s and Col Horton’s uncanny ability to teach modeling and game theory this thesis never would have happened. I truly appreciate their expert instruction and tutelage. Finally, I’d like to thank my wife, Linda, and children for their patience and support during this process.

- Chad

I would like to thank Dr. Giordano and Col Horton. Their unabashed enthusiasm for this subject and their ability to teach “Modeling for Dummies” led to this project. Their patience and insight made it flourish.

I would also like to thank my wife, Dee. She read every word of this thesis and made sure it stayed true to the original intent. As with our life, she made it better.

- Doug

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

> Do not worry about your difficulties in Mathematics. I can assure you mine are still greater.
>
> **- Albert Einstein**

Welcome to Modeling for Special Operations! Most of you are probably wondering why you're taking a math course for your Special Ops degree. Well, these courses are more about decision-making than math. In fact, let's not even call it math, let's just say "quantitative decision-making." You're going to learn some great tools that will help you make informed decisions after you graduate. You'll also learn to spot the errors when someone brings you a model. Okay, there is some math involved, but your instructor and this workbook will get you through. Don't worry if you haven't dusted off your algebra in some time. We've included memory joggers where necessary.

This workbook is not designed to replace your modeling textbook or your instructor. It is simply another tool to help you understand the math involved in the modeling concepts you'll see throughout your coursework. The workbook presents modeling lessons in a simple, step-wise fashion that guides you through the process of producing each model. It attempts, through the use of "layman's" language, to simplify the concepts and alleviate any math anxiety you may have. We hope you find it helpful. Now let's get into modeling.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LESSON 1 – MODELING CHANGE WITH DIFFERENCE EQUATIONS

What exactly is a model? One simple definition is that a model is a representation of reality. Of course, any representation will be incomplete and the same is true of all models you will see in your courses here. There is just no way to put all of reality into a model. Good models, though, provide valuable insights into a situation and allow you to come to reasonably good conclusions about it.

To come to these conclusions, we sometimes want to know what a value will be in the future. A simple way to model this future value is

$$\textit{future value} = \textit{present value} + \textit{change}$$

We take what we know the value to be now, add some change value and we will know what the future value will be. But, where does the change value come from? Well, we'll get this value from observing several instances of the reality we're trying to model. We'll see the change during these observations and we can use it to model further change. We can determine the change value by reworking the future value model from above.

$$\textit{change} = \textit{future value} - \textit{present value}$$

This says the change value is equal to the difference between a future value and the present value. So, we're predicting the future now? No, not yet, but we are talking about differences over time. Another way to think of it is that the amount of change is equal to the difference between the values at two successive time periods for whatever system we're trying to model, like the change in the balance of a savings account from one month to the next.

Okay, let's look at a mathematical expression of our verbal change equation from before. This equation models the change value and is called a difference equation. This is a general equation that we will use extensively in the first several lessons. Here it is

$$\Delta a_n = a_{n+1} - a_n$$

This equation reads "delta *a* sub *n* is equal to *a* sub *n* plus 1 minus *a* sub *n*." That's a mouthful and sounds like a brain full, but it is simply a mathematical statement of the verbal equation above that says the change in the value of *a* is equal to the future value minus the present value. Since it's a mathematical equation, let's look into it a little more. We'll start with the basic expression a_n (*a* sub *n*) then we'll look at the others. a_n is the present value of *a*. The subscript *n* represents any of the discrete time periods at which the value of *a* is taken, e.g., the first month or the second year. We could have a_1, representing the value of *a* at the end of the first time period, or a_2, representing the value of *a* at the end of the second time period, or a_{100}, representing the value of it at the end of the one hundredth time period, but the sub *n* represents any time period, the *n*th time period. *a* sub *n* plus 1 (a_{n+1}) then is the value at the end of the next time period, the value at the *n*th time period plus 1 or in our verbal equation, the future value. Delta *a* sub *n* (Δa_n) represents the amount of change at the end of the *n*th time period. The Δ (delta) means change. Remember, the value of *n* will be the same throughout the whole equation. Let's put in some numbers to help illustrate.

$$\Delta a_1 = a_2 - a_1$$

$$\Delta a_5 = a_6 - a_5$$

Great! Now let's look at a real world example.

Suppose you have a $1000 savings account that earns 1% interest per month. The amount of change over time can be modeled as a simple expression of the value at a given time multiplied by the interest. So, for your account, our change is equal to 1 percent of the value of *a* at a given time. Write it like this:

$$\Delta a_n = 0.01a_n$$

Since we know from our difference equation that

$$\Delta a_n = a_{n+1} - a_n$$

we can substitute and get

$$a_{n+1} - a_n = 0.01a_n$$

This equation allows us to predict future values of our account. We now need to set it up to solve for the future value (a_{n+1}.) Using a little bit of basic algebra (add a_n to both sides to get our future value by itself), we arrive at a model to predict future values of our account

$$a_{n+1} = a_n + 0.01a_n$$

This equation simply says that the future value (a_{n+1}) is equal to the present value (a_n) plus 1% of the present value. Our model is almost ready.

We need only one more element to have a complete model. That element is a_0. Since we are modeling change in value over time, it is important to know where we started. a_0 represents the initial value of our system, in our case, your account. The future value equation and the initial value together make a complete model called a dynamical system model. For your account the dynamical system model is

$$a_{n+1} = a_n + 0.01a_n,\ a_0 = 1000$$

With this model you have all the information you need to determine the value of *a* for each successive period of time; each *n*. To calculate the future value of *a* now, in this case a_1, just substitute the initial value, a_0, for the present value, a_n, and solve for a_{n+1}, the future value. It should look like this

$$a_{n+1} = 1000 + 0.01(1000)$$
$$a_{n+1} = 1000 + 10$$
$$a_{n+1} = 1010$$

Now that we have the value of a_1, we can calculate the value of a_2 then a_3 and so on. Our results give us a table that looks like this

n	a_n
0	1000
1	1010
2	1020.10
3	1030.30

Using these simple difference equations, the art of modeling becomes determining what to use for the change function. What do we add to, subtract from, multiply by, or divide into the present value to get the future value? Fortunately, it's usually fairly easy to determine the change function. Let's look at another example.

Suppose you bought a house 5 years ago. You financed $100,000 for 30 years at a monthly interest rate of .5%. Your monthly payments are $599.55. You won the

California lottery and want to pay off your loan so you need to know how much you still owe. Again, the art is determining the change function. In this case, the change in the amount you owe each period is a function of an increase (the interest) and a decrease (the monthly payment.) Write it like this

$$\Delta b_n = 0.005 b_n - 599.55$$

Substituting from the difference equation

$$\Delta b_n = b_{n+1} - b_n$$

we get

$$b_{n+1} - b_n = 0.005 b_n - 599.55$$

Then we put this equation in a form that allows us to predict the future value, b_{n+1},

$$b_{n+1} = b_n + 0.005 b_n - 599.55$$

This equation simply says that the future value is equal to the present value plus 0.5% of the present value minus the monthly payment of $599.55. Add the initial value, b_0, and we arrive at the following dynamical system model

$$b_{n+1} = b_n + 0.005 b_n - 599.55,\ b_0 = 100,000$$

where b_n is the amount owed after n months. Since you have made payments for 60 months, you would need start at b_0 and calculate out to b_{60} to determine how much you currently owe. Once again, start by substituting b_0 for b_n then solve for b_{n+1}. It should look like this

$$b_{n+1} = 100,000 + 0.005(100,000) - 599.55$$
$$b_{n+1} = 100,000 + 500 - 599.55$$
$$b_{n+1} = 99,900.45$$

Calculating out to $n = 60$ we get a table that looks like this

n	b_n
0	100000
1	99900.45
2	99800.40
3	99699.85
.	.
.	.
60	93054.39

Since we just want to get the concept of difference equations down right now, we won’t need to calculate all the values of our system at this time. In the near future, however, we’ll learn how to quickly plug these equations into Microsoft® Excel© and let it do the work for us.

That’s it. Remember difference equations allow you to model change taking place over discrete time periods. These equations allow you to model behavior exactly as it will occur over these time periods. Next lesson, we’ll look at using difference equations to approximate change that occurs continuously.

THIS PAGE INTENTIONALLY LEFT BLANK

III. LESSON 2 – APPROXIMATING CHANGE WITH DIFFERENCE EQUATIONS

In the last lesson we looked at change that occurred at discrete time periods—each month or each year. Unfortunately for us, most change does not cooperate to make modeling easier. Many changes are continuous. The change in temperature in your beer as it sits on the bar is a good example. The temperature doesn't change at discrete time intervals; it doesn't go up 2^{o} at each 5 minute interval. Instead it continuously rises as time goes by. We could use calculus to model this or we can approximate the continuous change using difference equations. We would prefer not to do the calculus, so we'll approximate. This process of approximating continuous change with difference equations is an example of model simplification.

Before we begin, we need to review a couple of topics: proportionality and lines. Proportionality refers to the mathematical relationship between two variables or unknowns. Two variables, for example x and y, are proportional to each other if one is always a constant multiple of the other. In other words, if there is a constant relationship between the variables, e.g., x is always twice y or y is always one-third of x, they are proportional.

The mathematical expression would look like this

$$x = ky$$

where x and y are unknown variables and k is the constant. If x is always twice y then you would have

$$x = 2y$$

where 2 is the constant. Some of the changes we will look at in the future will be represented by proportionalities, population is a good example.

We also need to discuss lines. In order to determine the change function for our models, we may need to graph our observed data, look at the line formed by the data and determine a numeric change from the graph. Refer to Figure 1 as we review. A graph has two axes, the x and y axes. The x axis is the horizontal axis and the y axis is the vertical axis. The point at which the x and y axis cross is called the origin. Generally each axis will represent some range of values. We plot data on the graph by placing a dot at the intersection of the *y* coordinate and the *x* coordinate. This then becomes point (*x*,*y*). Lines are essentially just a collection of plotted points. The line segments connecting any two of these points have the same slope. The slope is represented by the variable *m* and measures of how much *y* changes for a unit change in *x*. We show that mathematically by using the formula

$$m = \frac{\Delta y}{\Delta x}$$

which we can rewrite as

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

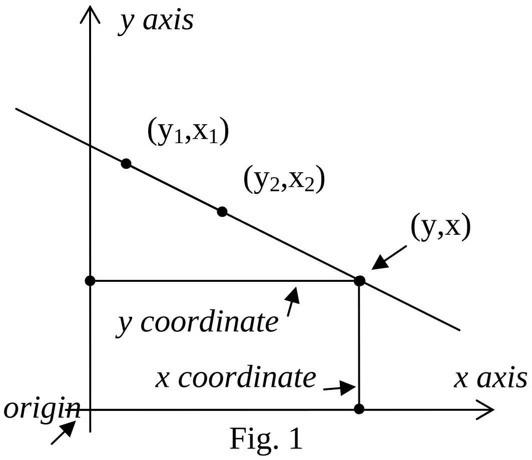


Fig. 1

where (x_1,y_1)and (x_2,y_2) represent any two sets of coordinates on our line. For example, if we have a line in which (x_1,y_1) = (1,6) and (x_2,y_2) = (3,5) we can determine the slope by substituting into our equation. This gives us

$$m = \frac{6-5}{1-3}$$

$$m = -\frac{1}{2}$$

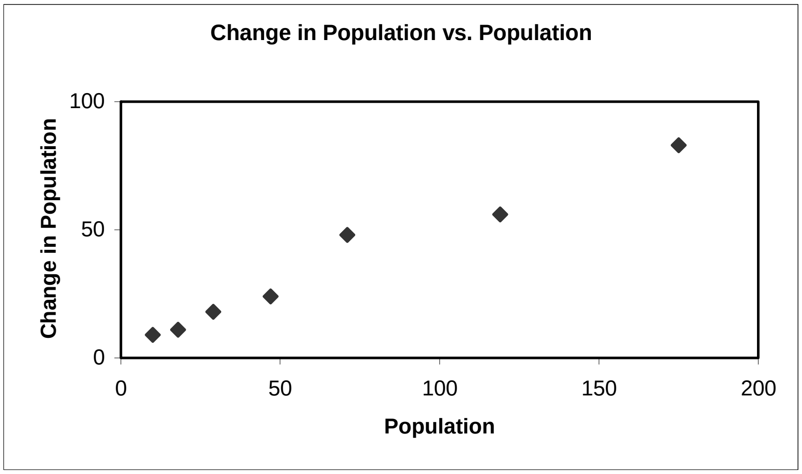
which says that every time the *x* coordinate changes by 2, the *y* coordinate will change by 1. The fact that the slope is negative tells us the line slopes down to the right. If it were positive we would know the line sloped up to the right. You can easily see that other coordinates for our imaginary line would be (4,5), (3,7), (2,9).

As we noted in the last lesson, the art of this whole thing is determining the change function. In this lesson, we'll graph our observations of our system and determine the relationship (proportionality) between the two variables by finding the slope of the line made by the graph of our data. The data will probably not cooperate to give us a perfectly straight line. However, as we said, we are going to approximate the change, so we will draw the line that best fits our data and use that to determine the relationship. Let's see a quick example.

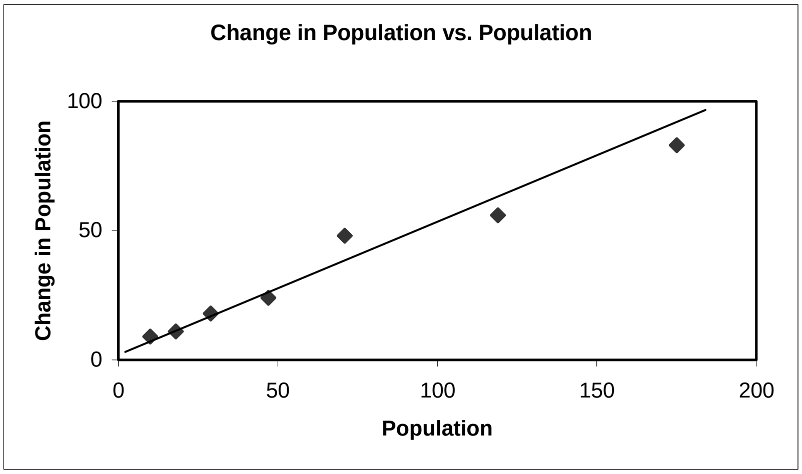
Suppose we want to model the population growth of a herd of animals. After collecting data, we come up with a chart of the growth that looks like this

Time in Months	Observed Population	Change in Population
0	10	8
1	18	11
2	29	18
3	47	24
4	71	48
5	119	56
6	175	83
7	258	

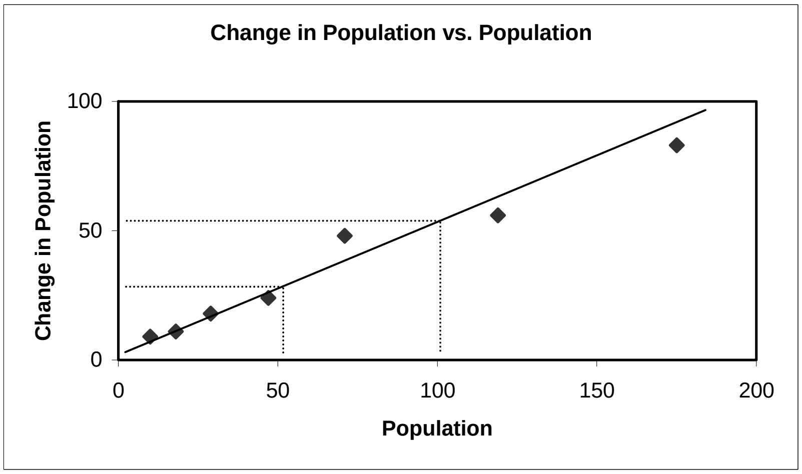
When we graph the change in population as compared to the population we get a graph that looks like this.



We can quickly see that this data does not lie on a straight line. So, we draw the best line we can through our data and we get a line that looks like this. Remember, we are approximating.



We'll use the slope of this line as the proportionality constant (*k*) in the relationship between the population and the change in the population. Once again, we're approximating the value of the *x* and *y* coordinates. Let's take a look at our graph.



We can see that when the value on the *x* axis is 50, the value on the *y* axis is about 25 and when the value on the *x* axis is 100, the value on the *y* axis is about 52. We plug these values into the equation for the slope of a line, solve

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$m = \frac{52 - 25}{100 - 50}$$

$$m = \frac{27}{50}$$

$$m = 0.54$$

and we get a slope of about 0.5 or $\frac{1}{2}$ (remember, approximation). This tells us that when the population (the *x* coordinate) increases 2, the change in the population (the *y* coordinate) increases 1. In other words the change in population is equal to 0.5 of whatever the population is. If we use P_n to represent the population of the herd at the end of day *n*, we can write it like this

$$\Delta P_n = 0.5 P_n$$

We can use the change function to predict future values for our population. We use our difference equation from the last lesson,

$$\Delta P_n = P_{n+1} - P_n$$

substitute for ΔP_n and get

$$P_{n+1} - P_n = 0.5 P_n$$

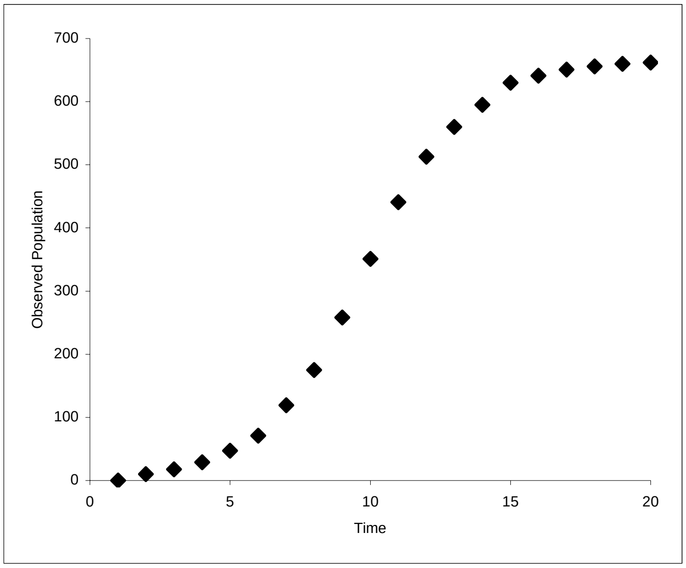
Then set up to solve for the future value (add P_n to both sides) and get

$$P_{n+1} = P_n + 0.5 P_n$$

This equation will approximate the future population numbers of our herd. Simple enough, but there is one problem. This equation infers that the population could grow indefinitely. Most systems, like our herd, exist in some constrained environment. That environment will only support so much growth. In other words there is some limit to the growth. Let’s refine our herd example.

A table of data on the herd’s growth in a restricted environment (limited food, limited area in which to live, etc.) and the resulting graph look like this:

Time in Months n	Observed Population P_n	Change in Population $P_{n+1} - P_n$
0	10	8
1	18	11
2	29	18
3	47	24
4	71	48
5	119	56
6	175	83
7	258	93
8	351	90
9	441	72
10	513	47
11	560	35
12	595	35
13	630	11
14	641	10
15	651	5
16	656	4
17	660	2
18	662	



We can see from the graph of our data that the growth of the population is slowing as the population gets bigger. We can reasonably guess that the population is approaching some limiting value or carrying capacity. In fact, most systems exist in an environment of limited resources that constrain the upper limit of the system.

What we need now is a model to approximate the change in population of our herd that takes the carrying capacity into account. Our difference equations have so far looked something like this

$$\Delta P_n = kP_n$$

where the change is some constant multiple of (proportional to) the current value. We used this for our last model of herd population growth. The problem was that this equation gives us unlimited growth and we know that the growth our population slows as it reaches some upper limit. Obviously we need something different in this equation to take into account the slowing population growth. Let's find out what we need by examining what we already know.

First, from our data, we can estimate the upper limit at about 665. Remember, we're approximating, so we could have used 664 or 666, but 665 is a nice round number so we'll use it. Next, we know that the growth at the limit will be zero. Expressed mathematically when P_n equals 665 ΔP_n must equal 0. We also know that as the difference between the number of animals in the herd and the limit (665) gets smaller, the rate of growth gets smaller. To illustrate, look at the table of observed data. When the difference between the population and limit is about 100 the growth is 35, but when the difference is about 30 the growth is only 11 and when the difference is 5 the growth is only 2. We must account for this decreasing growth in our difference equation. One way to do that is to multiply the difference between the limit and the population by the population, like this

$$(665 - P_n)P_n$$

Since we are subtracting an ever-increasing number as the population, P_n, gets larger or closer to the limit, the smaller the number the expression $665 - P_n$ represents. Therefore the expression $(665 - P_n)P_n$ yields a smaller number as the population grows. As a check, when P_n is 665, or the population limit, our expression $(665 - P_n)P_n$ equals 0. Therefore any change equation will equal 0 when P_n equals 0. This is exactly what we need from our model.

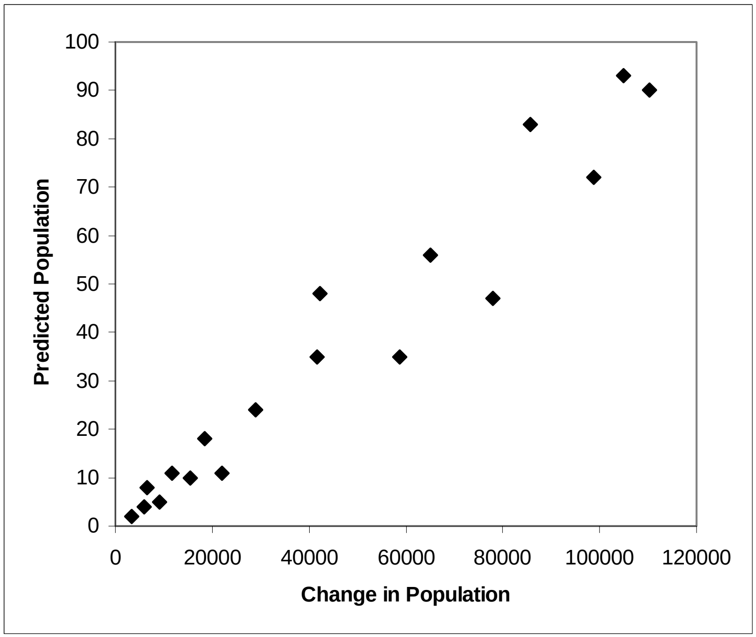
So, using that expression in our difference equation, we get

$$\Delta P_n = k(655 - P_n)P_n$$

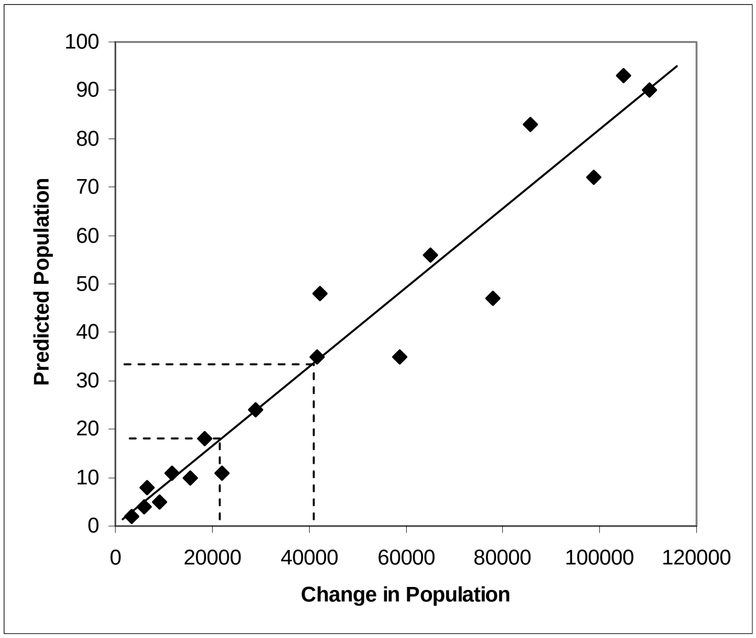
Now we just need to determine what our constant will be for this system. To do that we need to graph the data and determine the slope of the line that results. The slope of the line will be the relationship between the change and the population. First, let's put the data in a table

Change in Population $P_{n+1} - P_n$	Predicted Population $(665 - P_n)P_n$
8	6550
11	11646
18	18444
24	29046
48	42174
56	64974
83	85750
93	105006
90	110214
72	98784
47	77976
35	58800
35	41650
11	22050
10	15384
5	9114
4	5904
2	3300

Then we'll graph the change versus the predicted population



We can see that the data seems to originate at 0 and lie generally along a straight line. That's a good thing. Let's draw the best line we can and determine the slope.



Solving for the slope

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$
$$m = \frac{32 - 27}{40000 - 20000}$$
$$m = \frac{5}{20000}$$
$$m = 0.000025$$

we find it to be about 0.00025. We'll use this as our constant (*k*) in our difference equation, so the equation becomes

$$\Delta P_n = 0.00025(665 - P_n)P_n$$

Substituting for ΔP_n from the basic difference equation we get

$$P_{n+1} - P_n = 0.00025(665 - P_n)P_n$$

Rearranging to solve for the future values we get

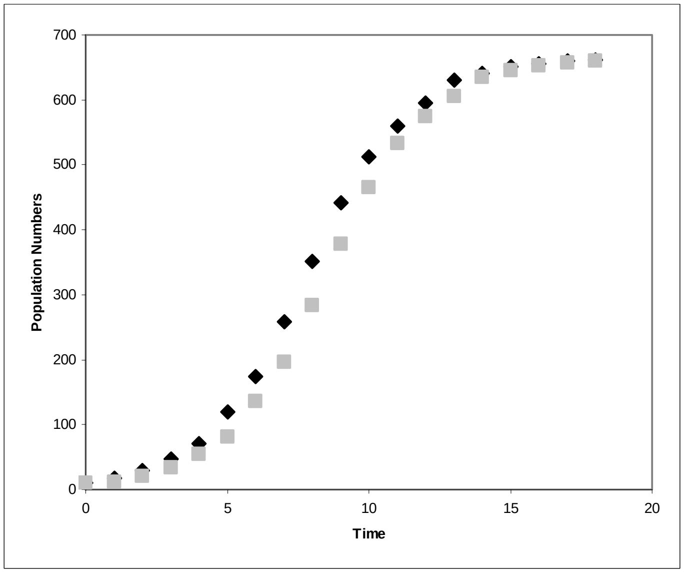
$$P_{n+1} = P_n + 0.00025(665 - P_n)P_n$$

Of course, this equation is no good without an initial value. From our observations we know $P_0 = 10$ so we can now generate predicted population numbers. We get a table that looks like this

Time in Months *n*	Predicted Population P_n
0	10
1	12
2	21
3	34
4	54
5	82
6	135
7	196
8	284
9	379
10	466
11	532
12	575
13	605
14	636
15	645
16	653
17	657
18	661

Let’s see how well we did with our predictions. Here’s table and resulting graph of the actual observed population numbers as compared to our predicted population figures.

Time in Months	Observed Population	Predicted Population
0	10	10
1	18	12
2	29	21
3	47	34
4	71	54
5	119	82
6	175	135
7	258	196
8	351	284
9	441	379
10	513	466
11	560	532
12	595	575
13	630	605
14	641	636
15	651	645
16	656	653
17	660	657
18	662	661



Great work! While our data doesn’t match the actual numbers perfectly, we did capture the trend of the data fairly well. In predicting the future, this would be a home run. Let’s look at one more example.

Suppose we need to spread some information through a village with a population of 500 people. We have some intel on the village and a contact who can give us various information regarding the culture and personal interactions within the village. Our boss wants to know when 75% of the village will have the information. Based on our contact's knowledge of the village, we can give our boss a reasonable answer. Let's see how.

First, let I_n represent the number of people in the village who have the information. If we assume everyone in the village is capable of obtaining the information, then those who don't have the information are represented by the total number of people in the village minus those people who have the information or $500 - I_n$. We must also assume that in order to pass the information along, those who know the information must interact with those who don't. We'll show this interaction as the product of the two, $I_n(500 - I_n)$. If every interaction between those who know and those who don't resulted in the information being passed, this expression would be our model. However, we know from our contact that only about a third of these interactions will actually result in the information being passed. So, based on the data we have, we can reasonably predict that about a third of the interactions between those who know the information and those who don't will result in a change in the number of people who have the information. We can write this as a difference equation.

$$\Delta I_n = 0.33 * I_n(500 - I_n)$$

We would then go on to solve for future values of I_n using the same method we used in our other examples. A graph of our predictions would have the same S shape as in the previous example.

Well, that's it. Hopefully, we've seen that the equations are the easy part and the challenge, albeit a fairly easy one, is determining what the change function is. Luckily, we can gather all the information we need to make this determination. With a little bit more practice, we'll soon be predicting change with one arm tied behind our backs. Next lesson we'll look at modeling using systems of difference equations in which change in one system affects the change in another.

IV. LESSON 3 – SYSTEMS OF DIFFERENCE EQUATIONS

So far we've looked at difference equations to model change in relatively simple systems. Now let's apply this newly gained knowledge to more complex systems. The equations are essentially the same. Once again, though, it is the determination of what constitutes change that makes or breaks our models in these systems of difference equations. Let's jump right in.

Suppose you're in charge of motor pools for a medium-sized operation. You have two motor pools, let's call them Alpha and Bravo. To facilitate better flow of vehicles you allow customers to sign out vehicles at one motor pool and turn them back in at the other. You want to make sure you have enough vehicles in both motor pools to satisfy demand. You also want to know if you need to ship vehicles between motor pools. You need a model of your system.

After analyzing the vehicle records, you determine that 70% of the vehicles signed out from Alpha are turned in at Alpha, whereas 30% are turned in at Bravo. You also determine that 60% of the vehicles signed out of Bravo are turned in at Bravo and 40% are turned in at Alpha. So, how many vehicles are at either motor pool at the end of the day? Let's start with Alpha. From our vehicle records, we know that the number of vehicles at Alpha at the end of the day is equal to 70% of those at Alpha at the beginning of the day plus 40% of those at Bravo. Mathematically it looks like this.

$$A_{n+1} = 0.70A_n + 0.40B_n$$

We also know from the records that the number of vehicles at Bravo at the end of the day is equal to 30% of those at Alpha at the beginning of the day plus 60% of those at Bravo. The equation looks like this.

$$B_{n+1} = 0.30A_n + 0.60B_n$$

These two equations make up a model for our system. What can this model do for us?

Let's say you want to reduce the number of vehicles you have to transport back and forth, so you need to know how many vehicles you should start with at each location. More specifically, is there a starting number at each location that would eliminate the need to transport any vehicles? In other words, is there a number of vehicles you can start with at each motor pool so that after all checking out and signing in is done, the number of vehicles at each motor pool doesn't change? In modeling, we would call this an equilibrium value. Let's rework our equations just a bit. We want a perfect system; one without change. Therefore, instead of A_{n+1} or A_n, which in the past we have used in difference equations to denote change, let's just use *A* without the change, and instead of B_{n+1} or B_n we'll just use *B*. That gives us the following equations:

$$A = 0.70A + 0.40B$$
$$B = 0.30A + 0.60B$$

Since we have a fixed number of vehicles, we know that the number of vehicles at Alpha will be related to the number of vehicles at Bravo. We can solve for A relative to B by using the equation for B in place of the variable B in the equation for A, like this:

$$A = 0.70A + .040(0.30A + 0.60B)$$

Now we multiply 0.40 through the second factor and get

$$A = 0.70A + 0.12A + .024B$$

Then we combine like terms (the two values of *A* on the right side) and get

$$A = 0.82A + 0.24B$$

Now, we want A on one side of the equation and B on the other, so we need to eliminate 0.82A from the right side. To do this we need to subtract it from the right side. Of course, whatever we do to one side, we must do to the other side so we end up with (remember the variable A by itself is the same as $1A$)

$$A - 0.82A = 0.82A + 0.24B - 0.82A$$
$$0.18A = 0.24B$$

Then, to get *A* by itself, we divide by 0.18 (again, what we do to one side we do to the other) and we end up with

$$\frac{0.18A}{0.18} = \frac{0.24B}{0.18}$$

$$A = 1.33B$$

So, this means that in order for our system to be in equilibrium, the number of vehicles at Alpha must always be 1.33 times the number of vehicles at Bravo. Another way to look at it is that A must always be $\frac{1}{3}$ greater than B. Since we have 700 vehicles, if we start with 300 at Bravo and $\frac{1}{3}$ more or 400 at Alpha, we should be at the equilibrium value.

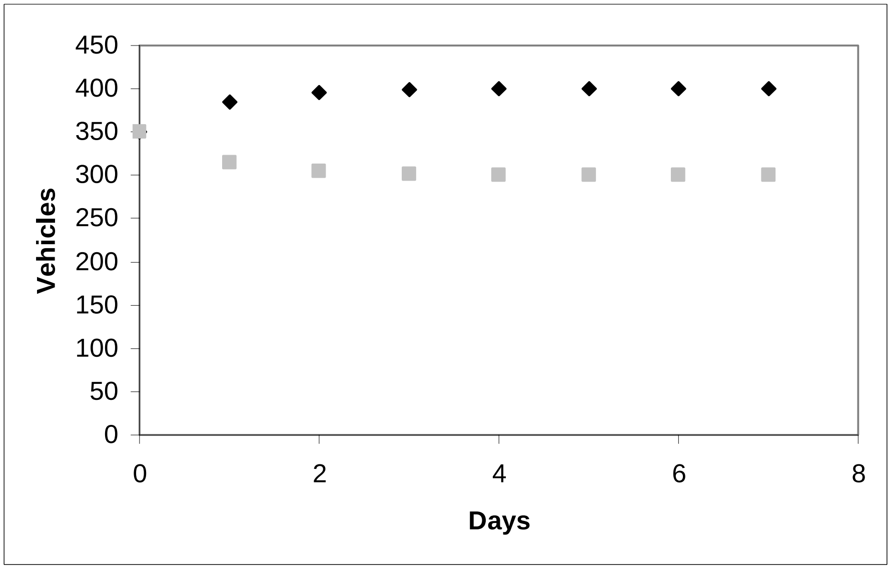
Let's plug in some numbers and see how they look. You have a total of 700 vehicles. Without any indications either way, you might start with 350 at each motor pool. What will the system do with that? Let's take a look. Here are the equations.

$$A_{n+1} = 0.70(350) + 0.40(350)$$
$$B_{n+1} = 0.30(350) + 0.60(350)$$

They give us the following table

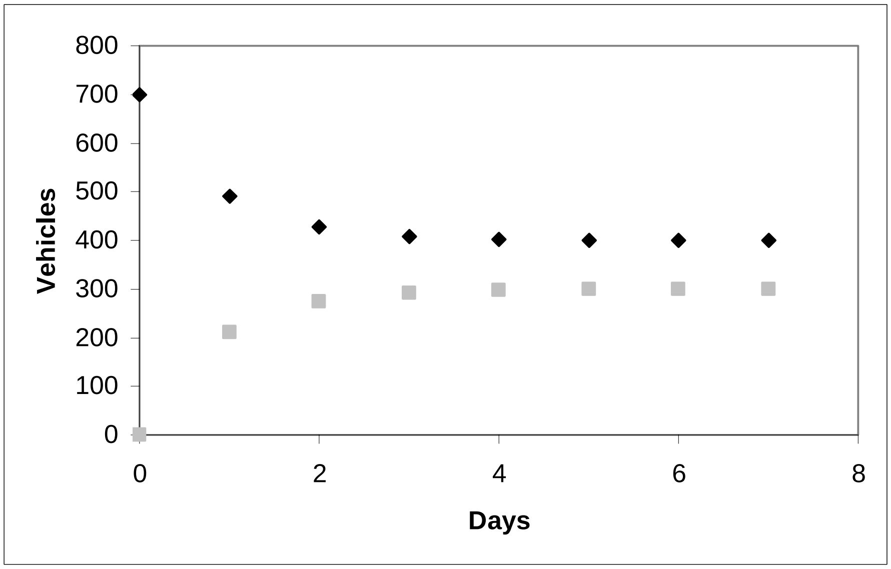
N	Alpha	Bravo
0	350	350
1	385	315
2	395.5	304.5
3	398.65	301.35
4	399.595	300.405
5	399.8785	300.1215
6	399.9636	300.0365
7	399.9891	300.0109

and the following graph.



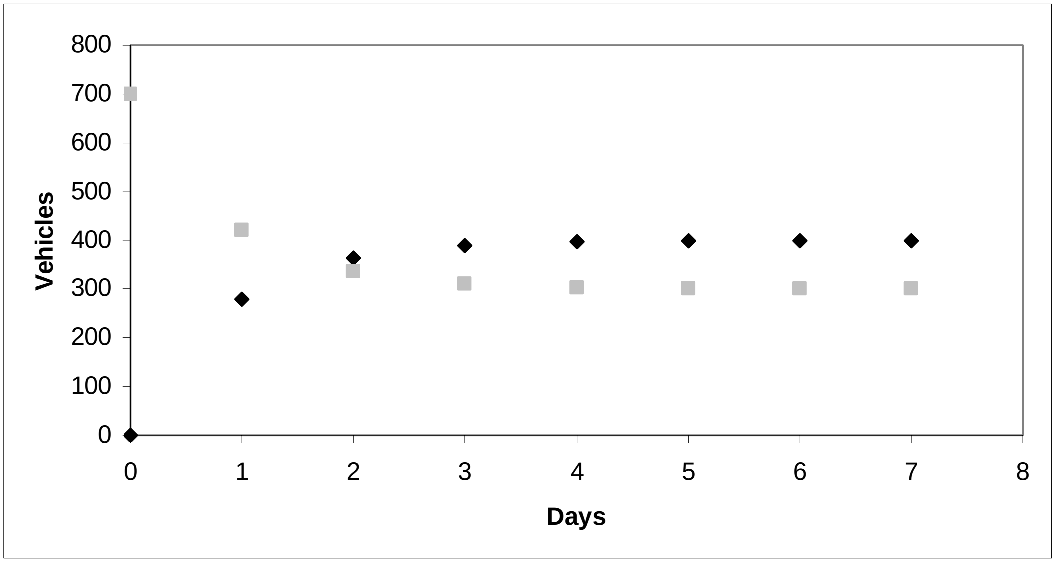
Interesting. The numbers seem to evening out at 400 vehicles at Alpha and 300 Vehicles at Bravo. What happens if we change the starting numbers, say start with all 700 vehicles at Alpha?

N	Alpha	Bravo
0	700	0
1	490	210
2	427	273
3	408.1	291.9
4	402.43	297.57
5	400.729	299.271
6	400.2187	299.7813
7	400.0656	299.9344



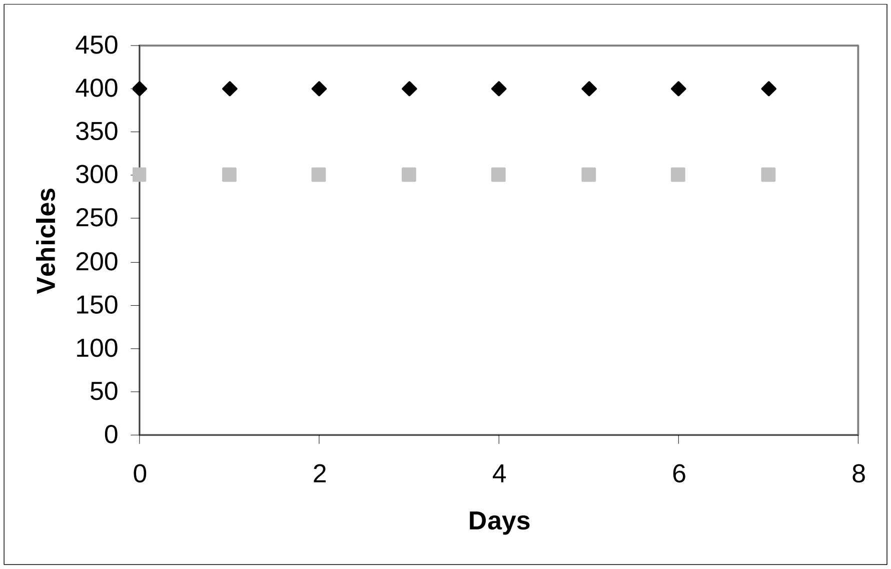
Even more interesting. Within about a week we end up with 400 vehicles at Alpha and 300 at Bravo. Let’s try one more. What if we start all the vehicles at Bravo?

n	Alpha	Bravo
0	0	700
1	280	420
2	364	336
3	389.2	310.8
4	396.76	303.24
5	399.028	300.972
6	399.7084	300.2916
7	399.9125	300.0875



Okay, now this is just weird . . . or is it? Actually, it's not. As we calculated earlier, this system has equilibrium values, values for which no change in the system takes place. If we start with 400 vehicles at Alpha and 300 at Bravo, the number of vehicles at both motor pools would not change.

n	Alpha	Bravo
0	400	300
1	400	300
2	400	300
3	400	300
4	400	300
5	400	300
6	400	300
7	400	300



400 vehicles at Alpha and 300 vehicles at Bravo, those are the equilibrium values of our system. No matter where we start, as long as the turn-in percentages remain the same, the equilibrium value will always be the same. We say that the equilibrium value of this system is stable because it stays the same, 400 and 300. We also say that it is insensitive to starting values because no matter what the starting values are, the system will always reach the same equilibrium. We'll look at systems in equilibrium in more detail in the next lesson. For now, let's look at another system of difference equations.

In 1805, at the Battle of Trafalgar, a British fleet of 27 ships under the command of Lord Nelson encountered combined French and Spanish fleet of 33 ships under Napoleon's control. During each stage of the battle each suffers losses equal to 10% of the number of ships of the opposing force. If we let B_n equal the number of British ships and F_n equal the number of French-Spanish ships at stage *n*, the change in the number of ships for each force looks like this mathematically.

$$B_{n+1} = B_n - 0.1F_n$$
$$F_{n+1} = F_n - 0.1B_n$$

This model yields the following data

Stage	British Force	French Force
0	27	33
1	23.700	30.300
2	20.670	27.930
3	17.877	25.863
4	15.291	24.075
5	12.883	22.546
6	10.629	21.258
7	8.503	20.195
8	6.483	19.345
9	4.549	18.696
10	2.679	18.242

We see from our data that the British are soundly defeated and limp home with 2 ships and a piece of another. These results seem intuitive given the French-Spanish force started with more ships and the attrition numbers for both forces were the same.

Those of you who know your history know that Lord Nelson led the British to victory at Trafalgar. He did so with a divide and conquer strategy. Napoleon's ships were arranged along a line in three groups. The groups, from left to right, had 17, 3, and 13 ships respectively. Nelson devised a strategy to attack the middle force of the three (we'll call it force A) with 13 of his ships, holding 14 in reserve. He planned to combine any ships surviving the first attack with the 14 in reserve and engage the force of 17 (we'll call it force B). Lastly, he would attack the remaining force of 13 (we'll call it force C) with all remaining ships. Assuming that each side loses 5% of the number of ships of the opposing force in each stage of each battle, we end up with the following model and results.

$$B_{n+1} = B_n - 0.05F_n$$
$$F_{n+1} = F_n - 0.05B_n$$

Battle A

Stage	British Force	French Force
0	13.000	3.000
1	12.850	2.350
2	12.733	1.708
3	12.647	1.071

Battle B

Stage	British Force	French Force
0	26.647	18.071
1	25.744	16.739
2	24.907	15.451
3	24.134	14.206
4	23.424	12.999
5	22.774	11.828
6	22.182	10.689
7	21.648	9.580
8	21.169	8.498
9	20.744	7.439
10	20.372	6.402
11	20.052	5.384
12	19.783	4.381
13	19.564	3.392
14	19.394	2.414
15	19.273	1.444

Battle C

Stage	British Force	French Force
0	19.273	14.444
1	18.551	13.480
2	17.877	12.553
3	17.250	11.659
4	16.667	10.796
5	16.127	9.963
6	15.629	9.157
7	15.171	8.375
8	14.752	7.617
9	14.371	6.879
10	14.027	6.161
11	13.719	5.459
12	13.446	4.773
13	13.208	4.101
14	13.003	3.441
15	12.830	2.791
16	12.691	2.149
17	12.584	1.514

The French – Spanish fleet is soundly defeated, limping home with one ship intact and one badly damaged. The British survive with 12 of their ships intact and one in need of repairs. The student of history will note the predictions based on our model are similar to what actually happened at the Battle of Trafalgar. The divide and conquer strategy was the only way for Lord Nelson to win. Let's continue to hone our change-predicting skills with one more example.

Suppose that in your area of operations (AO) there are two competing insurgencies. Suppose also that in the absence of the other insurgency (and of course your absence), each one would show unconstrained growth in which the change in the number of insurgents during a discrete time interval, let's use a day, is proportional to the number of insurgents at the beginning of the interval. If we use I_n to represent the population (size) of one insurgency at the end of day *n* and S_n to represent the competing insurgency population (we could have used any variable) then the following equations represent the change.

$$\Delta I_n = k_1 I_n \text{ and } \Delta S_n = k_2 S_n$$

In these equations k_1 and k_2 are the constant positive growth rates. (Remember, when two variables are proportional, they are some constant multiple of each other.)

Okay, the above equations represent the change in each insurgency's population without the other insurgency present. But, we have both insurgencies in the same AO and they are competing for the support of the same population. Therefore, we must assume interactions between the two are likely to decrease the population of each. Normally, we model interactions between groups as the product of the two groups so the interactions will look like this

$$I_n S_n$$

We can assume that the size of the decrease from these interactions is proportional (some constant multiple of) to the interactions so we'll need a constant multiplied in

$$k_3 I_n S_n$$

We can also assume that the proportion of the decrease will be different for each insurgency (one has better tactics, one has better weapons, etc.), so we'll need different constants for each interaction.

$$k_3 I_n S_n$$
$$k_4 I_n S_n$$

The resulting model, which says that the change in population of insurgency *I* at the end of time period *n* is proportional to the population at the beginning of *n* minus a proportion of the interactions between insurgency *I* and insurgency *S*, will look like this.

$$\Delta I_n = k_1 I_n - k_3 I_n S_n$$

We'll have a similar model and similar equation for the change in population of the other insurgency and it will look like this

$$\Delta S_n = k_2 S_n - k_4 I_n S_n$$

To complete our model, we first need to know what our constants are. As always, these are key to the change and will come from observation of the population. Let's say these same two insurgencies operate in another AO and you've been able to get some info from whomever is in charge over there. The data tells you that insurgency *I* grows at a rate of 15% of its population and that insurgency *S* grows at a rate of 20% of its population. The data also tells you that about 0.1% of the interactions between *I* and *S* result in a decrease in the population of *I* and about 0.2% result in a decrease in the population of *S*. Let's plug these values into our model.

$$\Delta I_n = 0.15 I_n - 0.001 I_n S_n$$
$$\Delta S_n = 0.2 S_n - 0.002 I_n S_n$$

Now let's use our basic difference equation, $\Delta X_n = X_{n+1} - X_n$, to substitute for the Δ terms to get

$$I_{n+1} - I_n = 0.15 I_n - 0.001 I_n S_n$$
$$S_{n+1} - S_n = 0.2 S_n - 0.002 I_n S_n$$

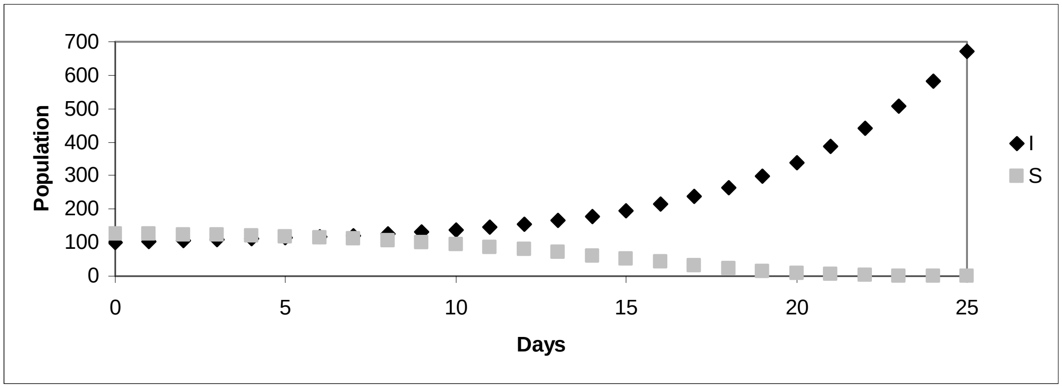
Then, solving for *I* and *S* at time $n+1$ we get

$$I_{n+1} = I_n + 0.15 I_n - 0.001 I_n S_n$$
$$S_{n+1} = S_n + 0.2 S_n - 0.002 I_n S_n$$

In plain English this model says that the population of each insurgency at the end of time period n , let's say a month, is equal to the population at the beginning of the month plus a percentage of the population at the beginning of the month (growth) minus a percentage of the interactions between itself and the competing insurgency (decline).

Let's count insurgents and see what our model tells us. You send out a recon patrol and find out that there are 100 insurgents in *I* and 125 in *S*. Plug these numbers into our model, solve, and we get the following data

n	I	S
0	100	125
1	102.5	125
2	105.0625	124.375
3	107.7547	123.1157
4	110.6516	121.2062
5	113.8377	118.6242
6	117.4095	115.3412
7	121.4787	111.3251
8	126.1769	106.5429
9	131.6602	100.965
10	138.1162	94.57181
11	145.7717	87.36238
12	154.9025	79.36493
13	165.844	70.65027
14	179.0037	61.34647
15	194.873	51.65327
16	214.0381	41.85227
17	237.1859	32.30676
18	265.1011	23.44269
19	298.6515	15.70187
20	338.7599	9.463467
21	386.368	4.944474
22	442.4128	2.112596
23	507.8401	0.665836
24	583.678	0.122727
25	671.1581	0.004006



We can quickly see from our model that insurgency *I* will grow enormously while insurgency *S* dies out altogether. This may be good and this may be bad. You might want both insurgencies to exist so they can deal with each other while you deal with each of them, making your job easier. Also, you don't want one insurgency growing out of control as in our example above. It may be that you want a balance among the insurgencies for one reason or another. You want to try to maintain a certain number of insurgents in each organization. You need an equilibrium value. Let's see if we can find one for our insurgency system.

First, because we want a system without change (change = 0), let's remove change from our model. As in the motor pool example above, we'll remove all the symbols for change (Δ, I_n, etc.) and use only the variable *I* or *S*. This results in the following

$$I = 0.15I - 0.001IS$$
$$S = 0.2S - 0.002IS$$

Then, because we want the change to equal zero, we need to set our equations equal to zero. We can then solve the equations to determine the value for each variable at which the change will equal zero. Like this

$$0 = 0.15I - 0.001IS$$
$$0 = 0.2S - 0.002IS$$

Since our equations each have two terms with the same variable (*I* in the first equation, *S* in the second), we'll first rewrite them

$$0 = I(0.15 - 0.001S)$$
$$0 = S(0.2 - 0.002I)$$

Now let's look at how to solve these equations for zero. Each equation has two terms; one, the variable outside the parentheses and two, all the numbers inside the parentheses. To ensure the equation equals zero, one of these two terms must be zero. Then, of course, anything multiplied by that term will result in the entire equation equaling zero. For the first equation, either the first term, *I*, can equal 0 or the second term, (0.15 – 0.001S), can equal zero. In order for the second term to equal 0, the variable *S* has to be equal to a certain value. We can find that value by setting this term equal to zero and solving for *S*, like this

$$0.15 - 0.001S = 0$$

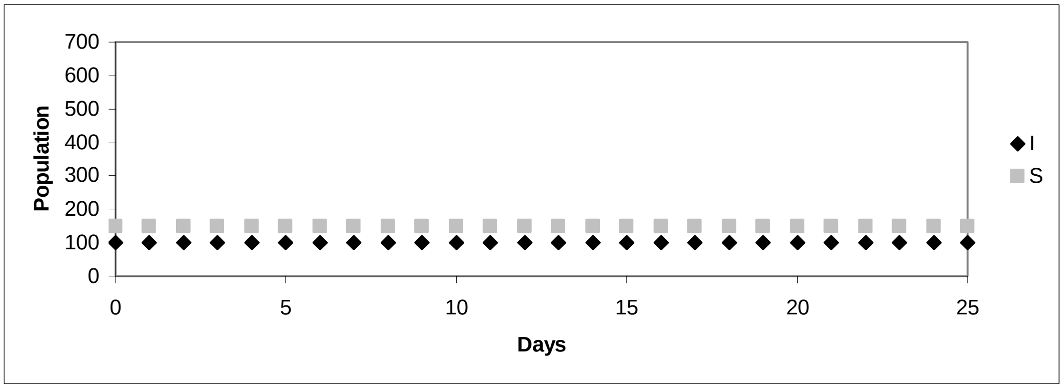
Then add 0.001*S* to both sides to get

$$0.15 = 0.001S$$

Now, to solve for *S*, divide both sides by 0.001 and we get

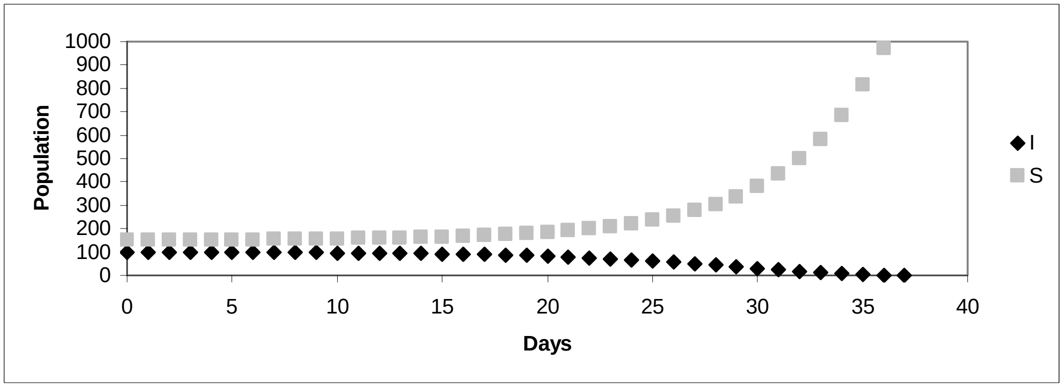
$$\frac{0.15}{0.001} = S$$
$$150 = S$$

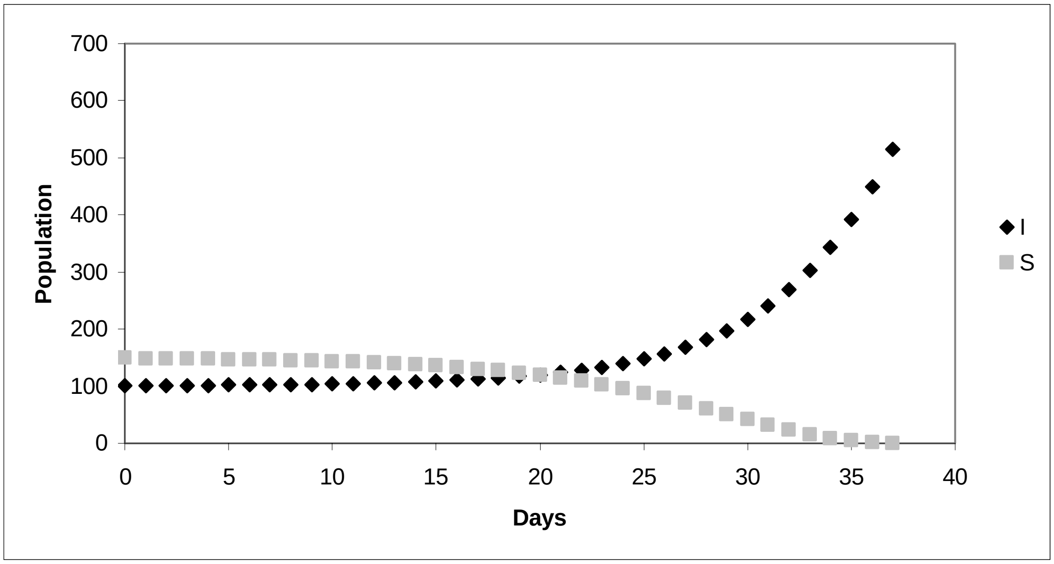
So, to sum up so far, our first equation equals zero, yields no change, when *I* equals zero or when *S* equals 150. We won't do the math here, but if we solve the other equation for zero also, we find that it yields no change when *S* equals zero or when *I* equals 100. This gives us the equilibrium values of our system. At either (*I*, *S*) = (0, 0) or (*I*, *S*) = (100, 150) there will be no change in the starting values. Let's plug those values into our model to see if we got it right.



Sure enough! 100 and 150 are the equilibrium values. If you can get the number of insurgents to 100 and 150, you could theoretically maintain this number indefinitely.

Now, let's take a look at the effect of minor changes in initial values on the equilibrium. Using starting values of (I, S) = (99, 151) and (101, 149) this is what the results look like





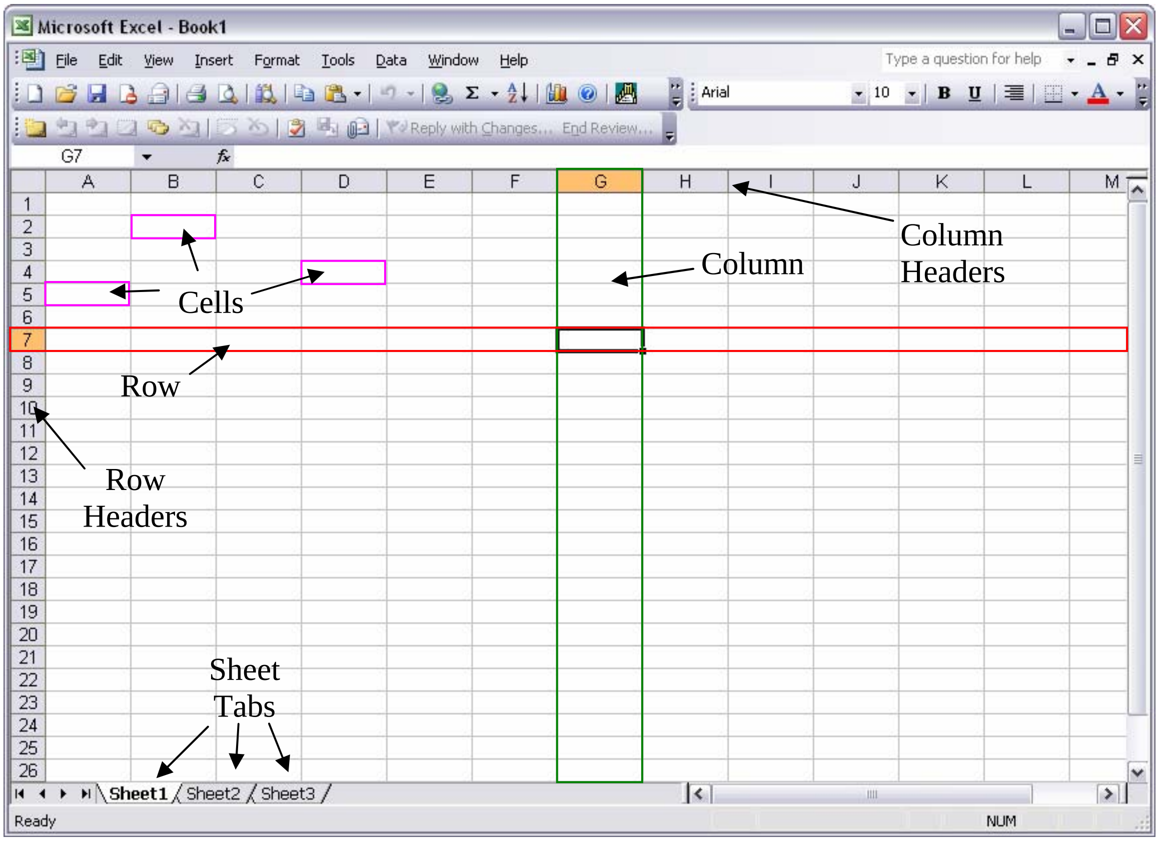
As you can see, if we start at any value other than the equilibrium, the populations swing wildly away from the equilibrium value. In modeling, we would say this system is extremely sensitive to initial conditions and is unstable because they will not assume equilibrium value even if we start very close. If you want to maintain the equilibrium value in this system, you must ensure that at some point the populations get to the equilibrium, they will then stay there. Remember though, the equilibrium is unstable, so in order to maintain it, you'll need to monitor the system for changing conditions such as a different growth rate or improved weapons that might change the attrition rate.

Okay, that's it for systems of difference equations. Remember, systems often contain several subsystems that have an effect on one another. To model the system, we must ensure we account for those effects. Next lesson, we'll learn how to use Microsoft® Excel© to do the math in these equations so we don't have to.

V. LESSON 4 – USING MICROSOFT EXCEL IN MODELING

Up to this point we have been working out the equations together, but you haven't had to calculate any of the number from these equations. Well, free time is over. This lesson we're going to look at how to write formulas in Excel to crunch these numbers for us. The equations have been pretty simple so far, but they will get a little more difficult and if we're going to use the data, we'd better know how to get it first. Let's jump right in.

The basic Excel screen is like most other Microsoft products. Menus and Toolbars at the top, scroll bars at the right and bottom, etc. You're probably pretty familiar with the interface. Excel has some distinct features, though. Refer to the figure below as we go through them. It might also be helpful to open up a copy of Excel on your computer to follow along.



When Excel opens it does so into a file called a Workbook. This is the basic name for an Excel product. The workbook is made up of a number of Sheets (the default is three), they're designated by the tabs at the bottom left of the screen. The sheet is made up of columns and rows. The intersection between a column and a row is called a cell. These are the rectangles you see all over the sheet and are the workhorse of Excel. Cells can contain data, formulas, text, and some other stuff we probably won't use. In the figure above, cell G7 is selected. You can tell because it has the heavy black border. You can also tell because the column and row headers for Column G and Row 7 are highlighted. Excel also tells you which cell you've selected by displaying its designator in the small window near the top left just below the toolbars.

To get to a certain cell in Excel, simply click on it. Pressing the TAB key moves you to the next cell to right. Generally pressing ENTER will move you to the cell below the one you're in. Excel thinks in blocks, so if you start at cell G7, do some work in cells H7 and I7 and press return, Excel figures you probably want to go back to enter more data in column G so it takes you to G8.

We're going to enter mostly formulas so Excel can crunch numbers for us. As far as Excel is concerned, it doesn't matter where you enter the data and formulas, just as long as you refer to the right cells when writing your formulas. Let's look at a quick example.

Back in Lesson 1, we determined a model for a simple interest bearing account. The model looked like this

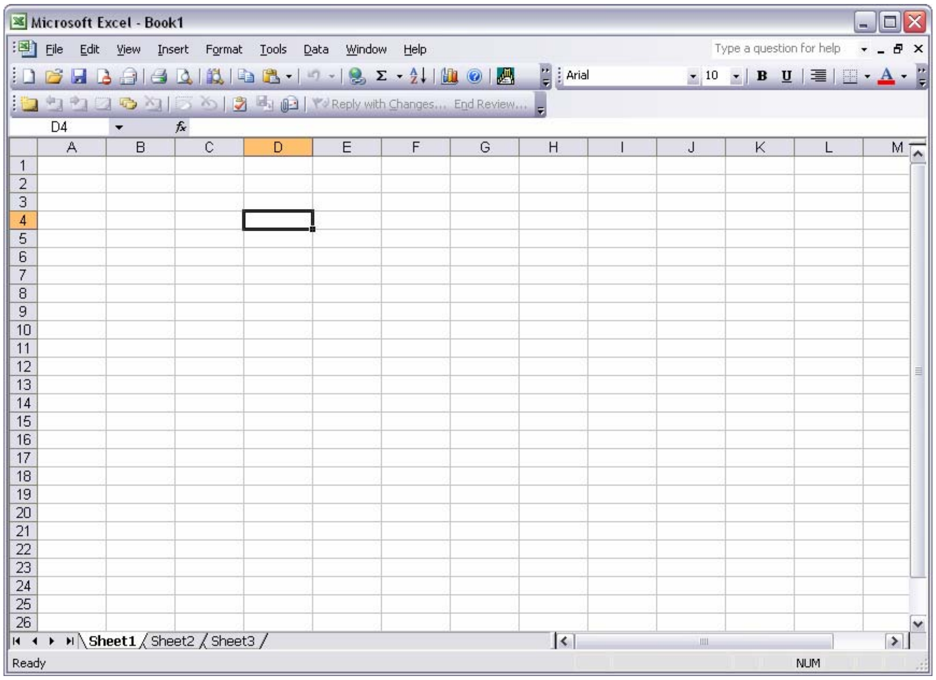
$$a_{n+1} = a_n + 0.01a_n,\ a_0 = 1000$$

We also developed a short table of data to show what the changes would look like based on our model. It looked like this

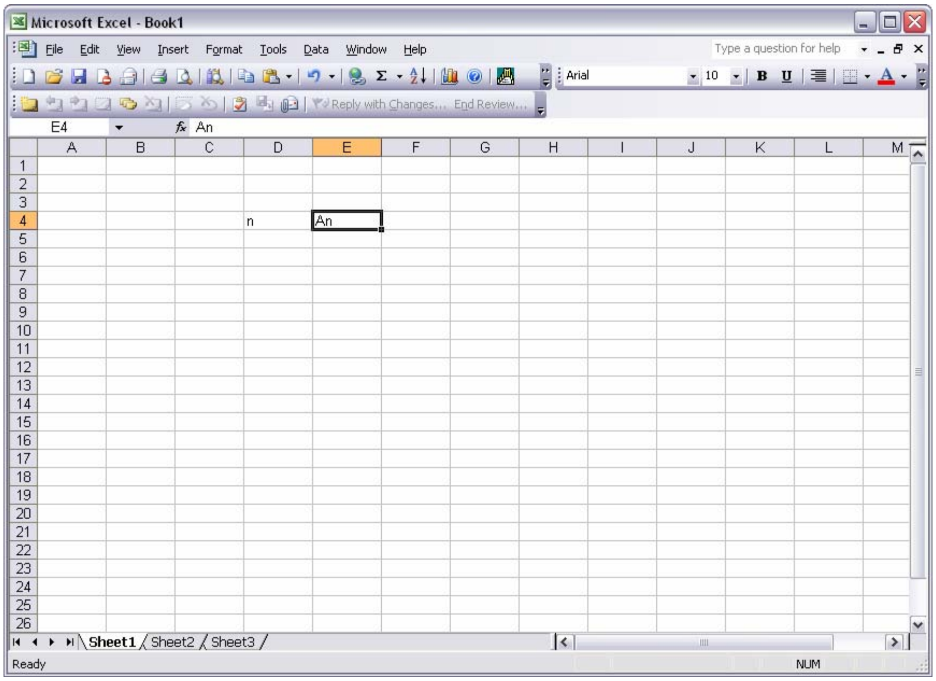
n	a_n
0	1000
1	1010
2	1020.10
3	1030.30

Now let's take a look at how to write the equation in Excel that will produce the data for our table.

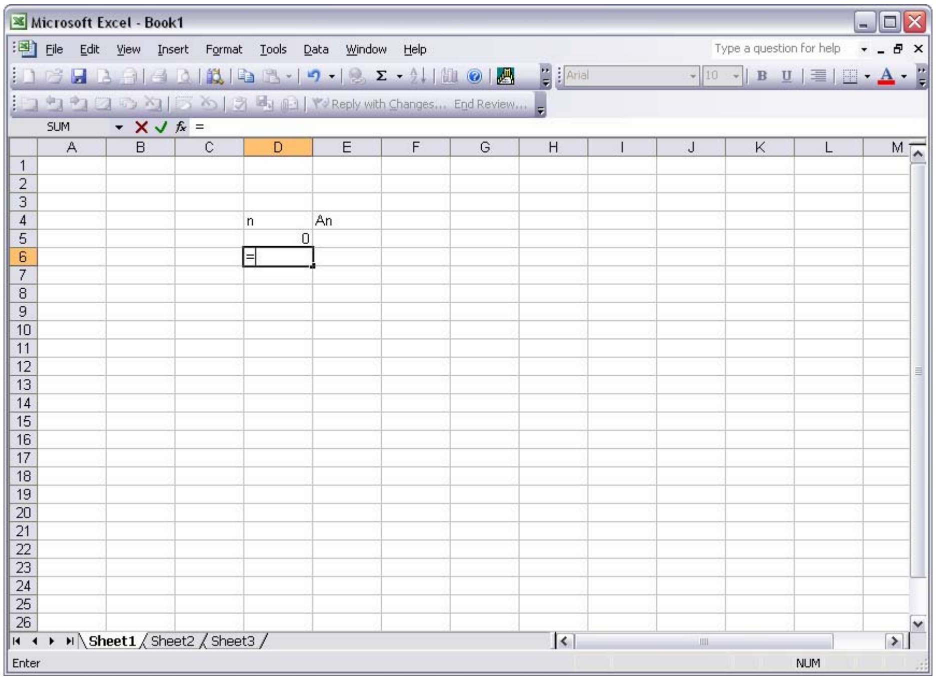
First open a blank workbook. Give yourself some room to work, select cell D4. Your workbook should look like this


Microsoft Excel - Book1
File Edit View Insert Format Tools Data Window Help
Type a question for help
Arial
10
Reply with Changes... End Review...
D4
A B C D E F G H I J K L M
Sheet1 Sheet2 Sheet3
Ready
NUM

Our data table in Excel will look a lot like the actual table we have so let’s put in our column headings. Type *n*, hit TAB and type *An*, your sheet should look like this


Microsoft Excel - Book1
File Edit View Insert Format Tools Data Window Help
Type a question for help
Arial
10
Reply with Changes... End Review...
E4
An
A B C D E F G H I J K L M
n
An
Sheet1 Sheet2 Sheet3
Ready
NUM

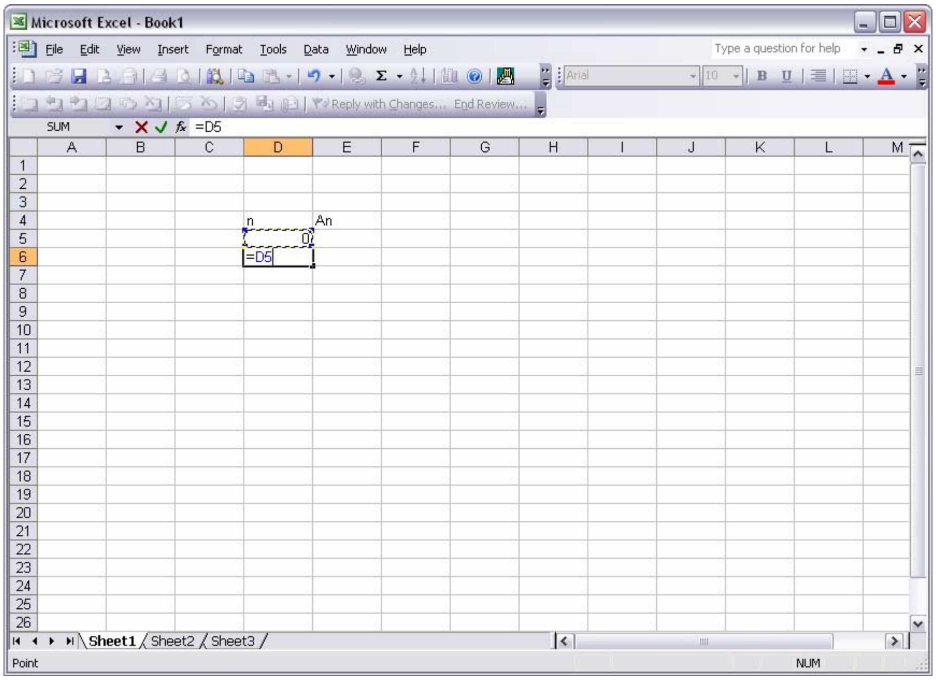
Now, we know we want the value of *n* to be 0, 1, 2, 3, etc. all the way down. You could type these values in one at a time, or you could write your first formula in Excel to do it for you. Simply type *0* in cell D5 then select cell D6. Now we’ll write the formula that tells Excel to take the value of the cell above, add one, and display it in the current cell. To do this, type an = sign, this tells Excel you want to write a formula. So far your sheet should look like this


Microsoft Excel - Book1
File Edit View Insert Format Tools Data Window Help
Type a question for help
Reply with Changes... End Review...
Arial
10
Sheet1 Sheet2 Sheet3
Enter
NUM

SUM =

	A	B	C	D	E
4				n	An
5				0	
6				=	

Now tell Excel the designation of the cell above. You can type in *D5* or you can select it. After doing this your sheet should look like this

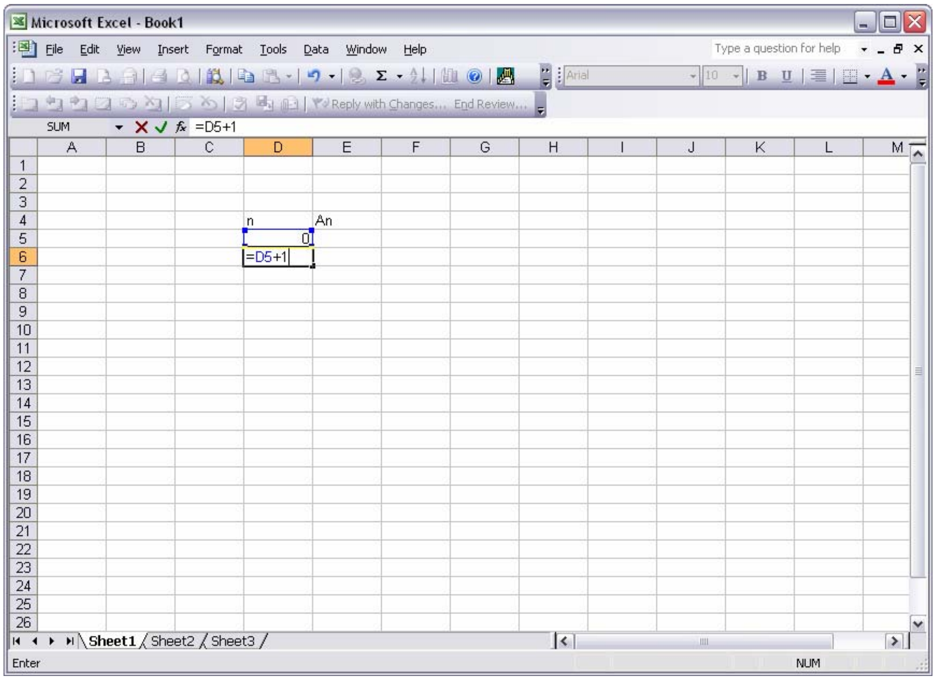

Microsoft Excel - Book1
File Edit View Insert Format Tools Data Window Help
Type a question for help
Reply with Changes... End Review...
Arial
10
Sheet1 Sheet2 Sheet3
Point
NUM

SUM =D5

	A	B	C	D	E
4				n	An
5				0	
6				=D5	

If you typed the cell designator, you'll have a solid blue border around the designated cell. If you selected the cell you want to use, it will have a sparkling border around it to show which cell you selected, as in the above figure.

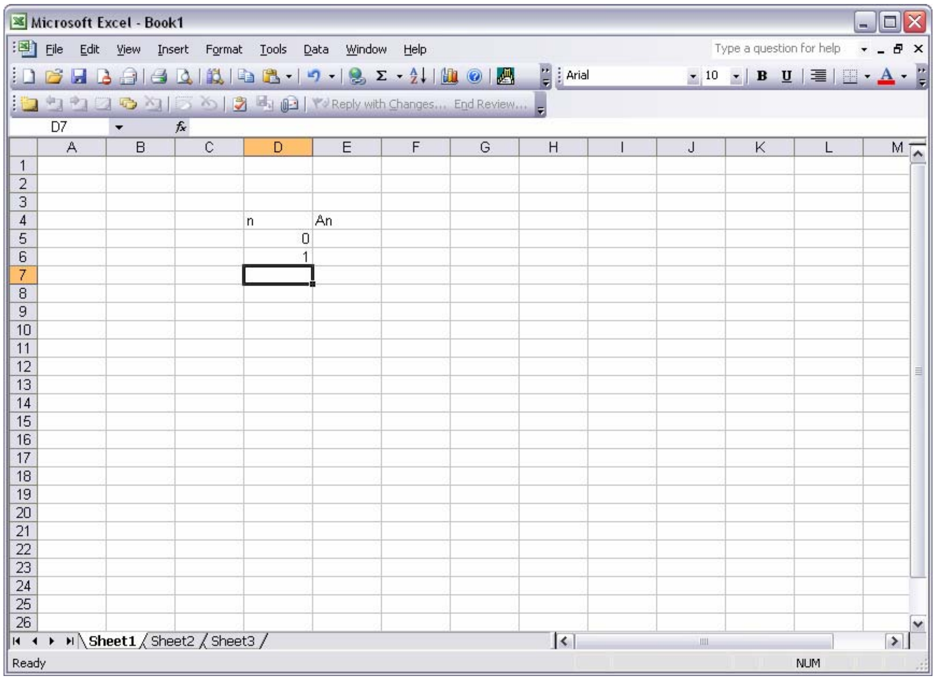
Now tell Excel you want to add 1 to the value of cell D5 by typing *+1*. Your sheet now looks like this.



SUM =D5+1

	D	E
4	n	An
5	0	
6	=D5+1	

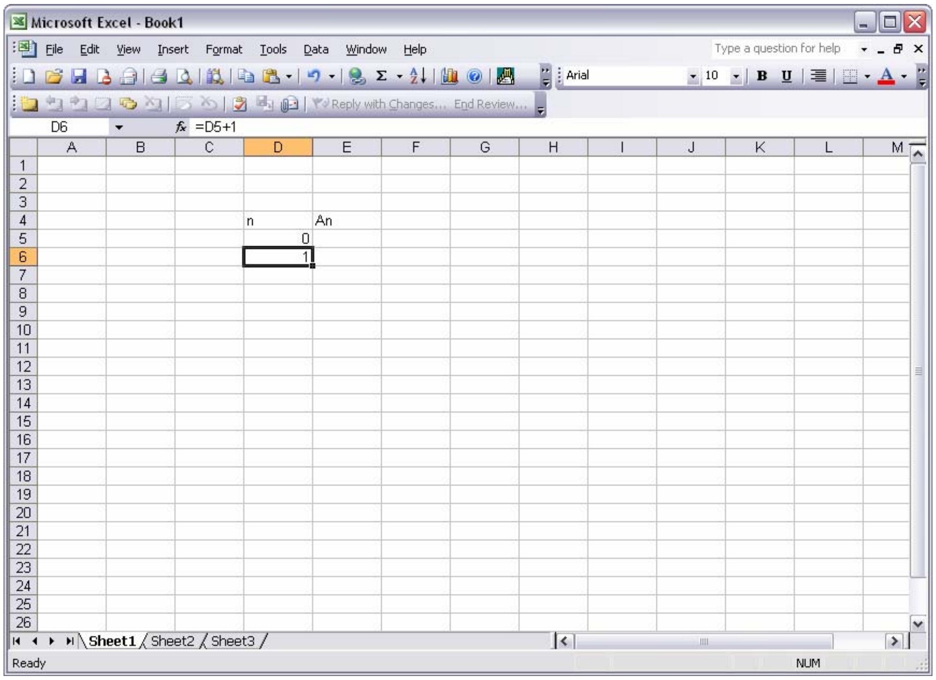
Hit enter and you've written your first formula. Congratulations! Your sheet should now look like this



Microsoft Excel - Book1
File Edit View Insert Format Tools Data Window Help
Type a question for help
Arial
10
Reply with Changes... End Review...
D7
Sheet1 Sheet2 Sheet3
Ready
NUM

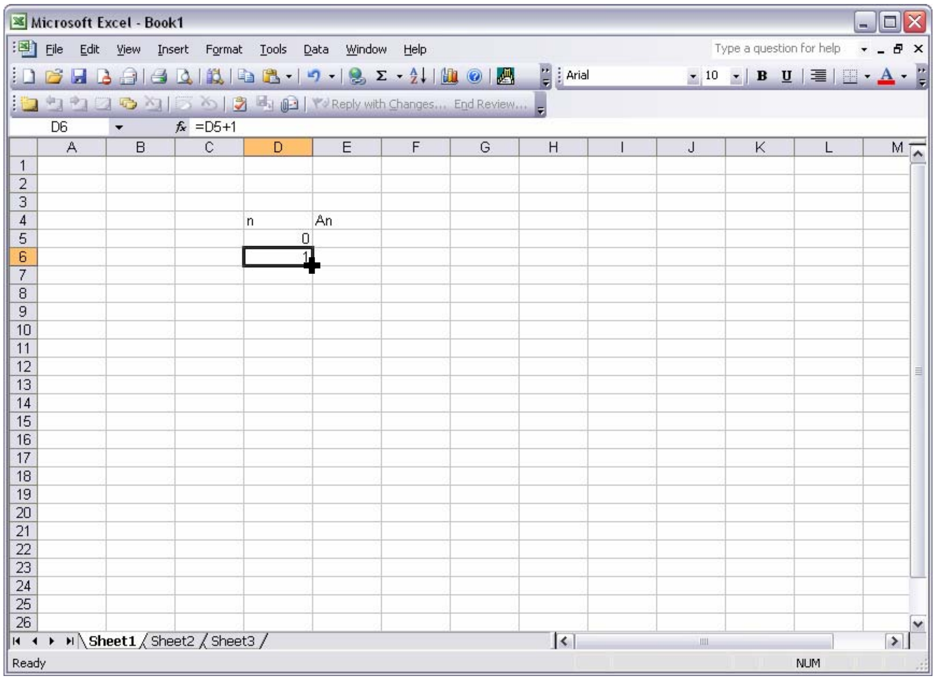
	D	E
4	n	An
5	0	
6	1	
7		

You can see that Excel has added 1 to the value of cell D5 to return a value of 1 in cell D6. Now we just need to write a similar formula in each cell below D6 until we have enough values for *n* to finish the problem. Fortunately Excel will do all of that for us. You can simply copy the contents of cell D6, your formula, into as many cells as you'd like. Do this by selecting cell D6, you'll notice a small square in the lower right hand corner. When you put the cursor over the square, it becomes a plus sign. Click and hold on that square and drag the mouse down until you think you have enough cells cover. For our example, copy the formula into 10 cells. The following four figures take you through the copying process.


Microsoft Excel - Book1
File Edit View Insert Format Tools Data Window Help
Type a question for help
Arial 10
Reply with Changes... End Review...
Sheet1 Sheet2 Sheet3
Ready
NUM

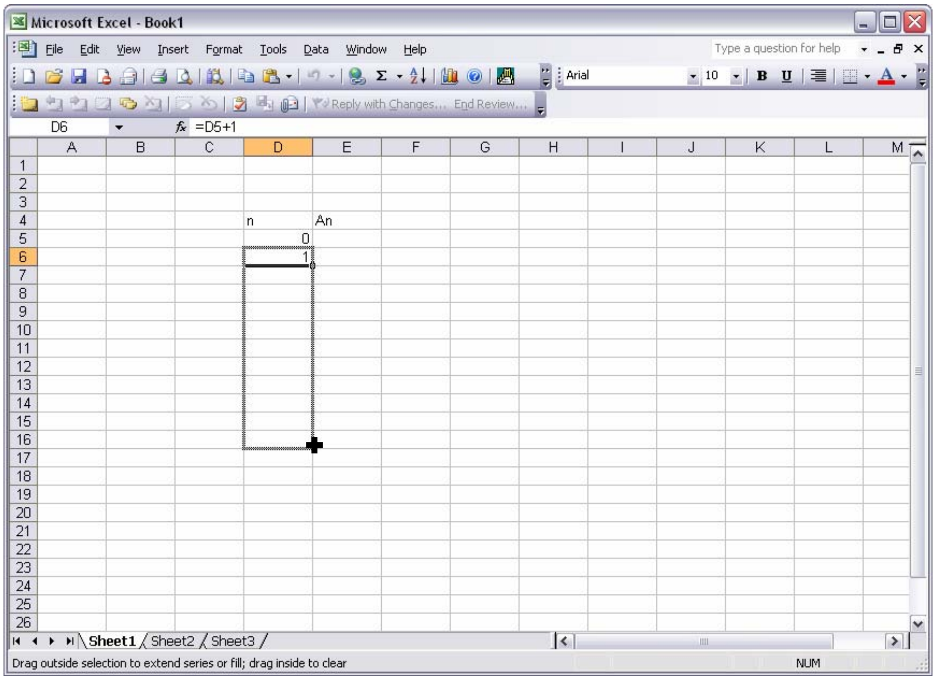
D6 =D5+1

	D	E
4	n	An
5	0	
6	1	


Microsoft Excel - Book1
File Edit View Insert Format Tools Data Window Help
Type a question for help
Arial 10
Reply with Changes... End Review...
Sheet1 Sheet2 Sheet3
Ready
NUM

D6 =D5+1

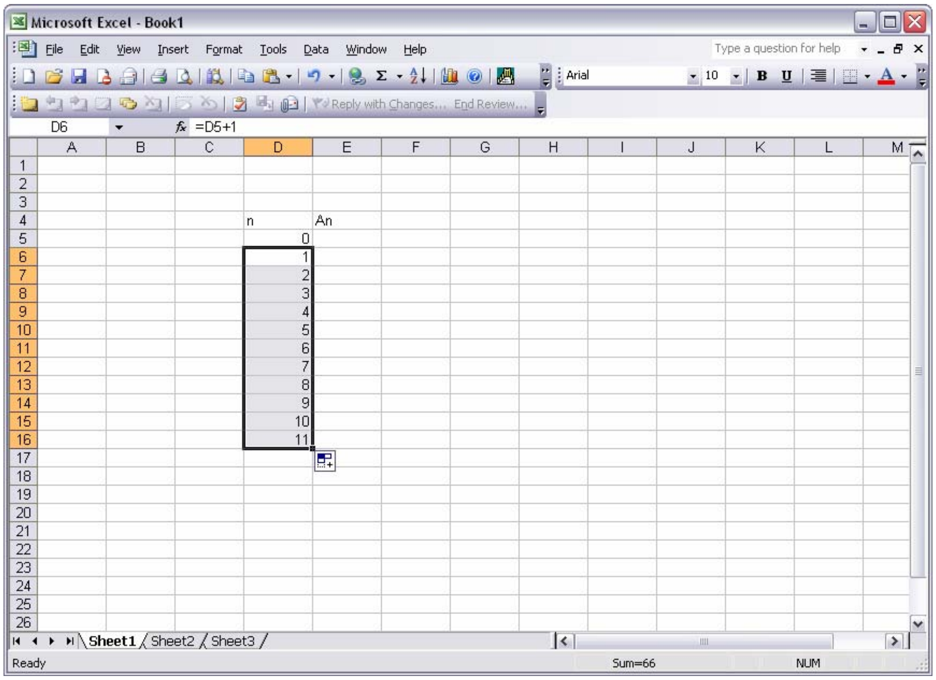
	D	E
4	n	An
5	0	
6	1	


Microsoft Excel - Book1
File Edit View Insert Format Tools Data Window Help
Type a question for help
Arial 10
Reply with Changes... End Review...
A B C D E F G H I J K L M
Sheet1 Sheet2 Sheet3
NUM

D6 =D5+1

	D	E
4	n	An
5	0	
6	1	

Drag outside selection to extend series or fill; drag inside to clear


Microsoft Excel - Book1
File Edit View Insert Format Tools Data Window Help
Type a question for help
Arial 10
Reply with Changes... End Review...
A B C D E F G H I J K L M
Sheet1 Sheet2 Sheet3
NUM

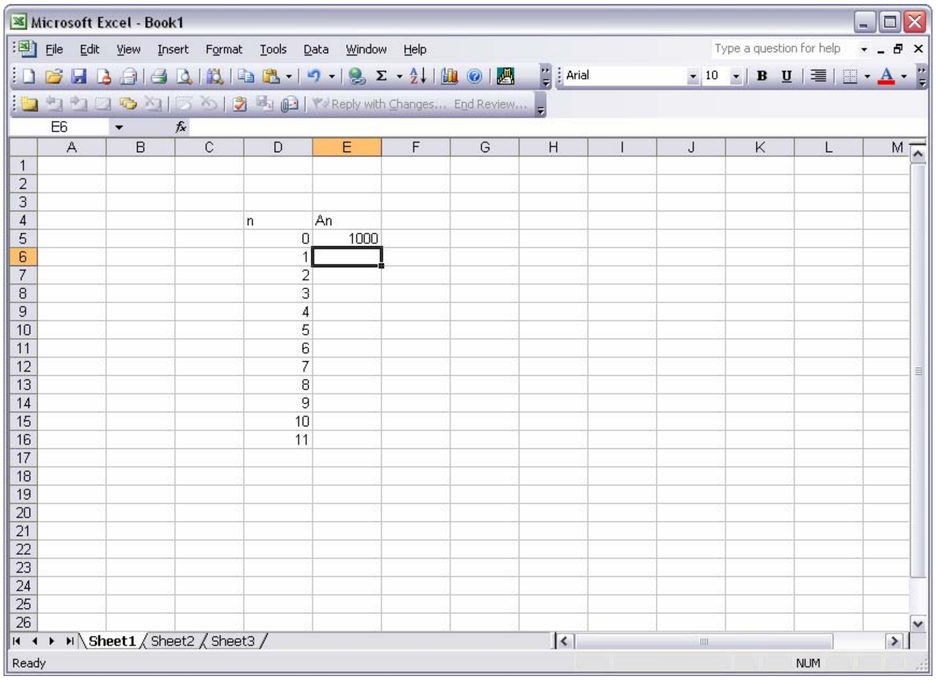
D6 =D5+1

	D	E
4	n	An
5	0	
6	1	
7	2	
8	3	
9	4	
10	5	
11	6	
12	7	
13	8	
14	9	
15	10	
16	11	

Ready Sum=66

Excel has created a column of formulas, each of which adds one to the value of the cell above it. Excel has a convenient feature of copying "relative" to the cells you're working with. Instead of making ten copies of cell D6, it copied formulas with relative cell designations. Look at the formula in D6, the one we wrote, and the formula in D7, the first one Excel copied for you. Notice the difference? The formula in D6 is D5+1, the formula in D7 is D6+1. Excel kept everything relative for you. There will be times when you don't want Excel to do this. We'll discuss how to tell Excel not to copy relative a little later.

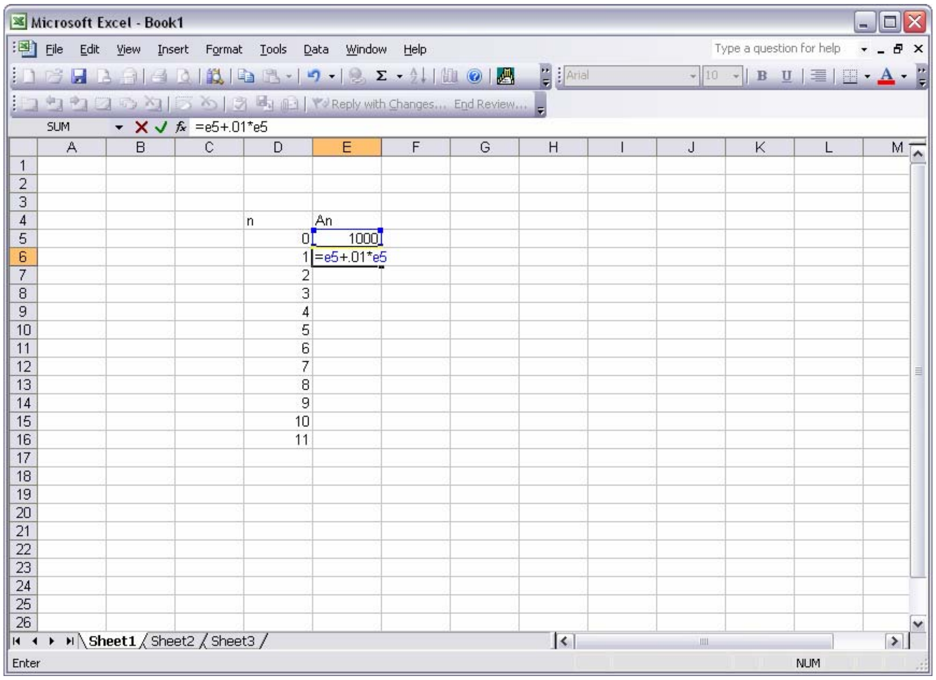
Alright, let's see how to write the formula that gives us our future values. First select cell E5. Since n = 0, this is where we'll put our initial value, A_0. Type *1000* and hit enter. Now you should have cell E6 selected, like this



Row	D	E
4	n	An
5	0	1000
6	1	
7	2	
8	3	
9	4	
10	5	
11	6	
12	7	
13	8	
14	9	
15	10	
16	11	

Okay, we know the equation for the future value of *A*, all we have to do is tell Excel what that formula is. We'll use some very basic symbols to tell Excel what calculation we want. All numbers go in as themselves, e.g. for 1 type *1*, for 243 type *243*. To add, type +, to subtract, type –. Multiplying and dividing are a bit different. To multiply type *, to divide type /. Excel follows mathematical rules regarding the order of operations; it does all operations inside parentheses, it multiplies then divides, adds then subtracts. Without any parentheses, Excel will just multiply, divide, add, and then subtract. If we want to do operations in a particular order, we can use parentheses to direct Excel to perform the functions in the order we want. Simply enclose the expression in parentheses, type *(* or *)*. We don't need to type =, Excel assumes you're done with your formula when you hit enter. Those are the basics. Let's write our formula.

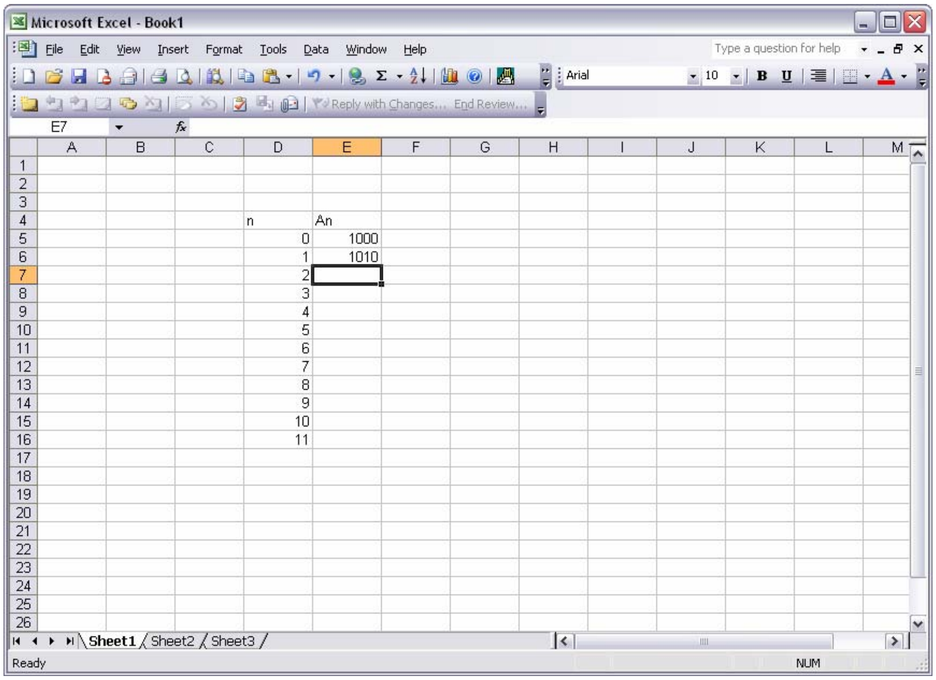
Start with the equals sign, =. That tells Excel we are writing a formula. Then type in a logical expression of our difference equation. Don't use numbers, unless they are constants. We want to use cell references whenever we can, so Excel will work for us. Our difference equation is $a_{n+1} = a_n + 0.01a_n$, so we want cell E6 to equal the value of E5 plus 0.01 times the value of E5. Again, we use cell references so Excel will copy relative for us, don't type in the values. We did type in *0.01* because this constant will be the same in all our cell calculations. So type the following into cell E6: *=e5+0.01*e5*. Remember, you can either type the designator for or select with your mouse the cell you want to reference. Either method will produce the right formula. You'll figure out which one works best for you. After typing your formula, your sheet should look like this


Microsoft Excel - Book1
File Edit View Insert Format Tools Data Window Help
Type a question for help
Arial
10
Reply with Changes... End Review...
Sheet1 Sheet2 Sheet3
Enter
NUM

SUM =e5+.01*e5

	A	B	C	D	E
4				n	An
5				0	1000
6				1	=e5+.01*e5
7				2	
8				3	
9				4	
10				5	
11				6	
12				7	
13				8	
14				9	
15				10	
16				11	

Notice that the contents of the cell appear in two places; in the cell where you're typing them and also in the formula bar at the top of the screen to the right of *fx*. You can edit your formulas in both places as well. You may find it easier to edit formulas in the formula bar. Okay, hit enter. Excel will do the calculation and give you the following

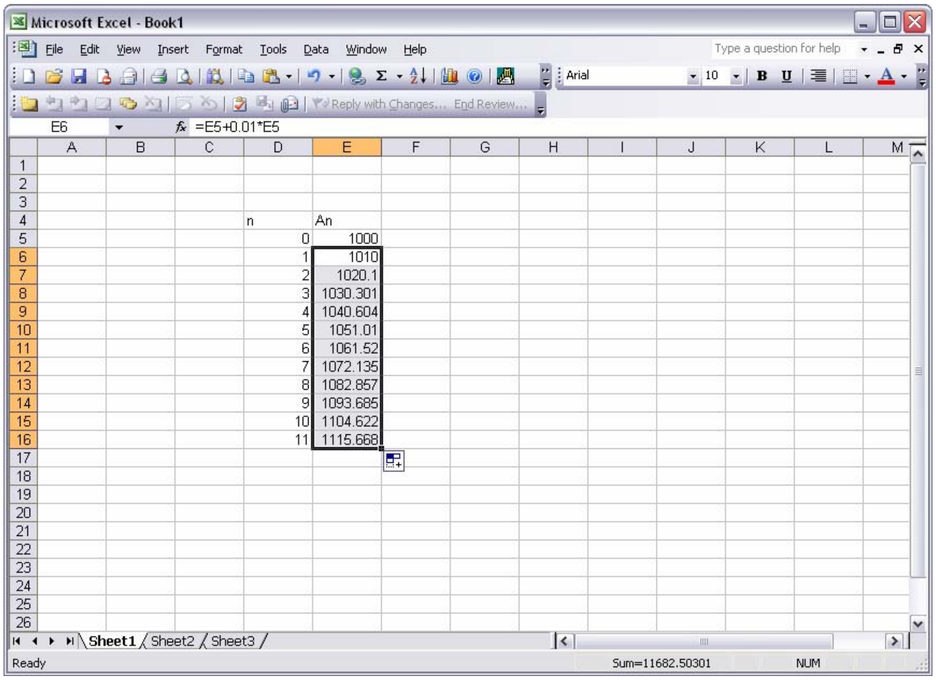


Microsoft Excel - Book1
File Edit View Insert Format Tools Data Window Help
Type a question for help
Arial
10
Reply with Changes... End Review...
E7
Sheet1 Sheet2 Sheet3
Ready
NUM

Row	D	E
4	n	An
5	0	1000
6	1	1010
7	2	
8	3	
9	4	
10	5	
11	6	
12	7	
13	8	
14	9	
15	10	
16	11	

If you’ve made some glaring error, Excel will tell you. If you’ve simply input bad data, it won’t. You’ll have to look at the result of your formula to see if it’s logical. If not recheck your formula.

Now we need to copy the formula in E6 down through E16. Just like before, select E6 then click and drag the small square in the bottom right hand corner of the cell down to E16. This will copy relative the formula in E6. Try it now and your sheet should look like this



E6 =E5+0.01*E5

	D	E
4	n	An
5	0	1000
6	1	1010
7	2	1020.1
8	3	1030.301
9	4	1040.604
10	5	1051.01
11	6	1061.52
12	7	1072.135
13	8	1082.857
14	9	1093.685
15	10	1104.622
16	11	1115.668

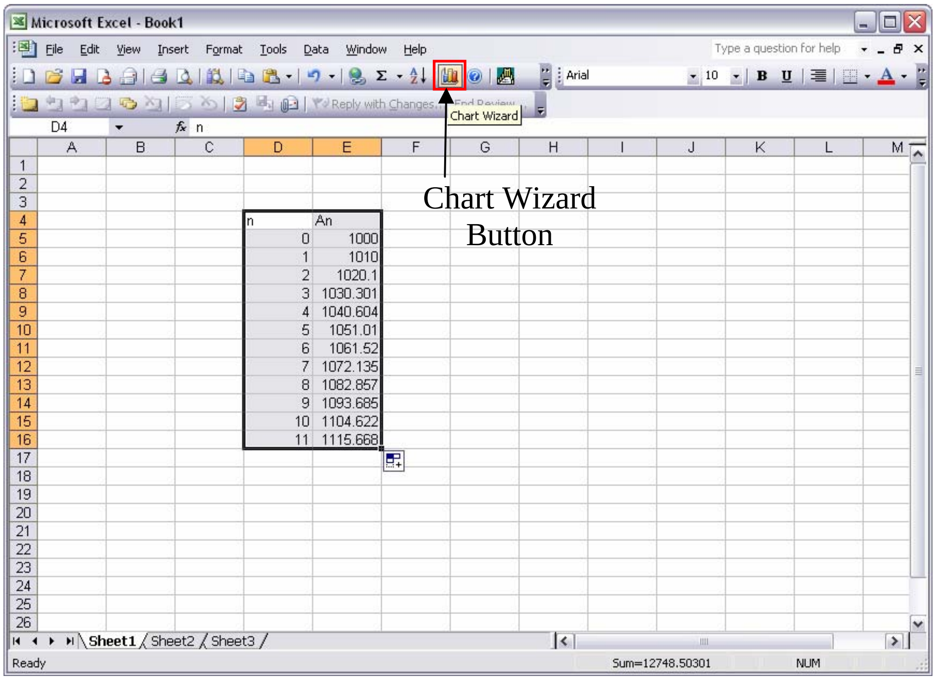
Ready Sum=11682.50301 NUM

Great work! We're modeling! If you've determined your change function correctly and input your formula correctly, Excel can quickly produce results for you. You'll impress your boss in no time.

Even more impressing though, is a good-looking graph of your data. We all know the boss just wants to see charts, right? Well with Excel, you can give him a great chart of your data, quickly. Let's see how.

Excel has a great feature called a chart wizard that let's you produce these graphs and charts with just a few mouse clicks. All you need to do is highlight the data you want to make a graph from, click the chart wizard button in the tool bar and follow a few simple prompts to get a great looking graph.

So, click and hold on the upper left cell of the data you want to graph. Now drag the mouse down to the lower right cell of your data and release. This should highlight your data. Now click on the Chart Wizard toolbar button



D4 = n

n	An
0	1000
1	1010
2	1020.1
3	1030.301
4	1040.604
5	1051.01
6	1061.52
7	1072.135
8	1082.857
9	1093.685
10	1104.622
11	1115.668

Sum=12748.50301

When the wizard opens you should get a dialog box that looks like this

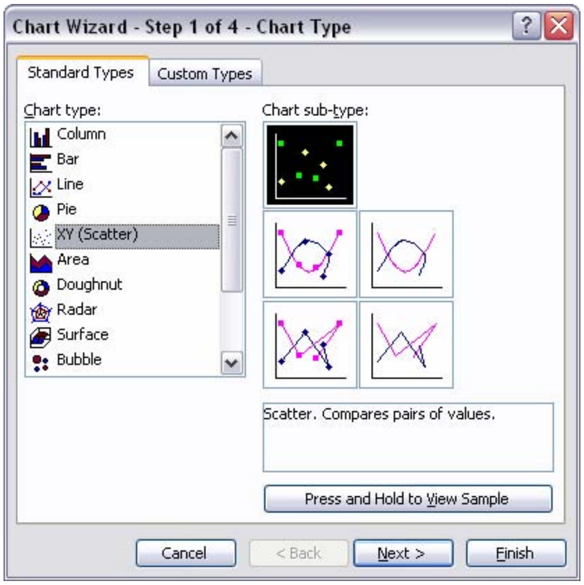
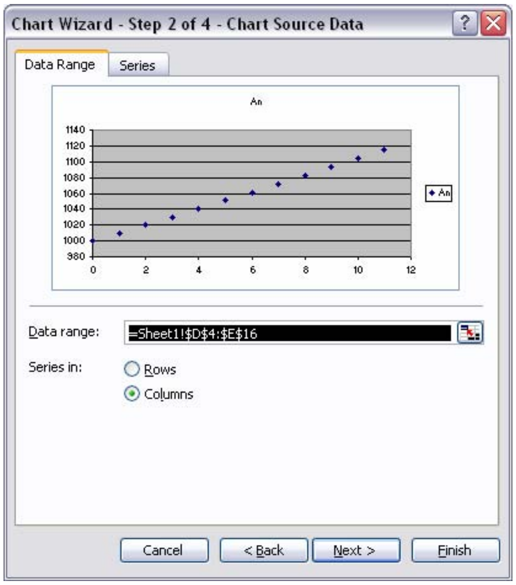


Chart type: XY (Scatter)

Scatter. Compares pairs of values.

Step 1 of the wizard asks you to choose the type of chart you want. You can play around with the different types, but for our purposes the XY (Scatter) chart fits the bill. Select it and press the Next > button.



Data range: =Sheet1!$D$4:$E$16

Series in: ○ Rows ● Columns

The next step asks you to choose the data range for your chart. If you selected the right data before opening the chart wizard, you shouldn't have to do anything at this step. Press the Next > button.

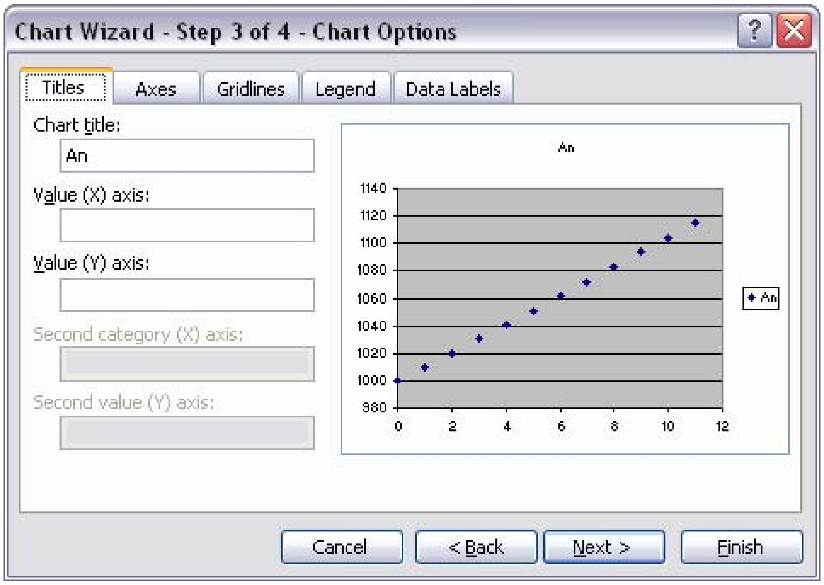


Chart title: An

Value (X) axis:

Value (Y) axis:

Second category (X) axis:

Second value (Y) axis:

This step allows you to add titles to your chart. You can change the title of the chart and add labels to your axes. If you're using this chart for the boss, you probably want to add some titles. For our purposes, the default settings are fine. Press the Next > button.

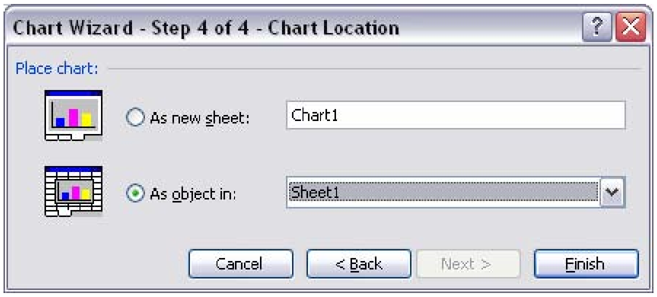


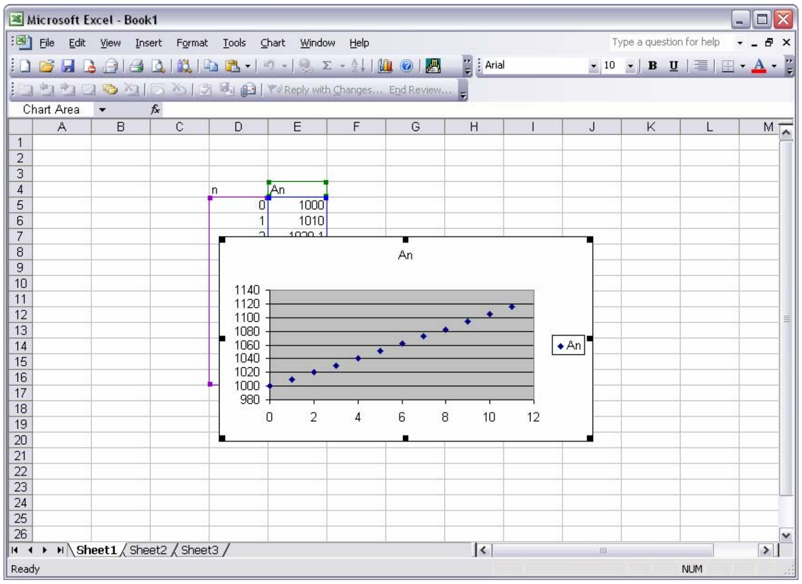
Chart Wizard - Step 4 of 4 - Chart Location

Place chart:

○ As new sheet: Chart1

● As object in: Sheet1

This last step let's you pick whether your chart should go on the current sheet, a different sheet, or a new sheet. If you have a lot of data on the current sheet, you may want to put your chart on another sheet. For most of our applications, it will be fine on the current sheet. Press the Finish button.



There you have it. A nice-looking chart giving a quick graphic representation of your data. As you can see, these charts allow a quick look at the trend of the data, something your boss will definitely be interested in.

You may have noticed while we were running the chart wizard we changed options only at step one. For the majority of our models, this will be true. Highlight your data, open the chart wizard, select the XY (Scatter) chart and press the Finish button. You'll have a nice chart in a matter of seconds.

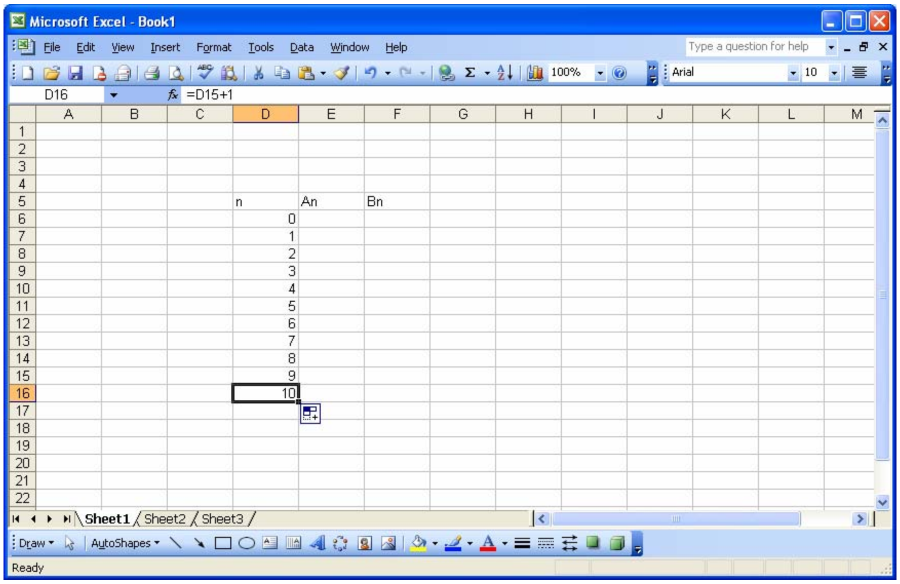
Let's take a look at one more technique before we finish up. We can write formulas in Excel that reference other cells. We'll use that feature to make our job easier, here's how.

We'll use the motor pool example from the last lesson to demonstrate this technique. Here are the equations

$$A_{n+1} = 0.70A_n + 0.40B_n$$
$$B_{n+1} = 0.30A_n + 0.60B_n$$

We discussed a little about the equilibrium in this model and we adjusted the starting values. We want to put data into Excel in such a way that we can freely adjust starting values (or other values) without deleting a formula or somehow messing up our hard work. To keep our formulas and other data safe, we'll put the values that we may want to change later in different cells and then have Excel refer to those cells in our formulas. Here's how.

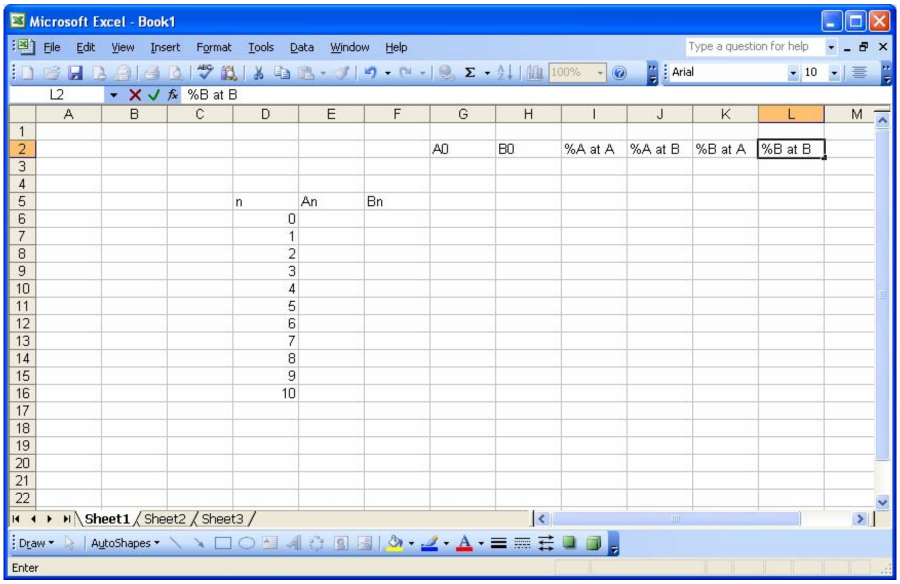
Using the same formulas as above set up your sheet with about 10 values for *n*, a column for A*n* and a column for B*n*. When you're finished, your sheet should look like this.



D16 =D15+1

	D	E	F
5	n	An	Bn
6	0		
7	1		
8	2		
9	3		
10	4		
11	5		
12	6		
13	7		
14	8		
15	9		
16	10		

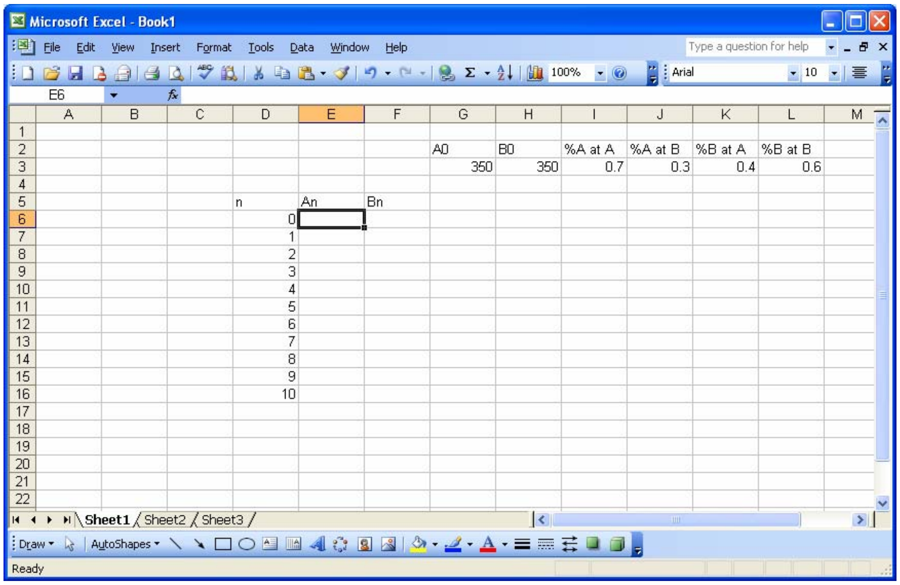
Now we'll put in some other column headings to the right of our working columns to act as holding places for our values. Something like this



L2 %B at B

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2							A0	B0	%A at A	%A at B	%B at A	%B at B	
3													
4													
5				n	An	Bn							
6				0									
7				1									
8				2									
9				3									
10				4									
11				5									
12				6									
13				7									
14				8									
15				9									
16				10									
17													
18													
19													
20													
21													
22													

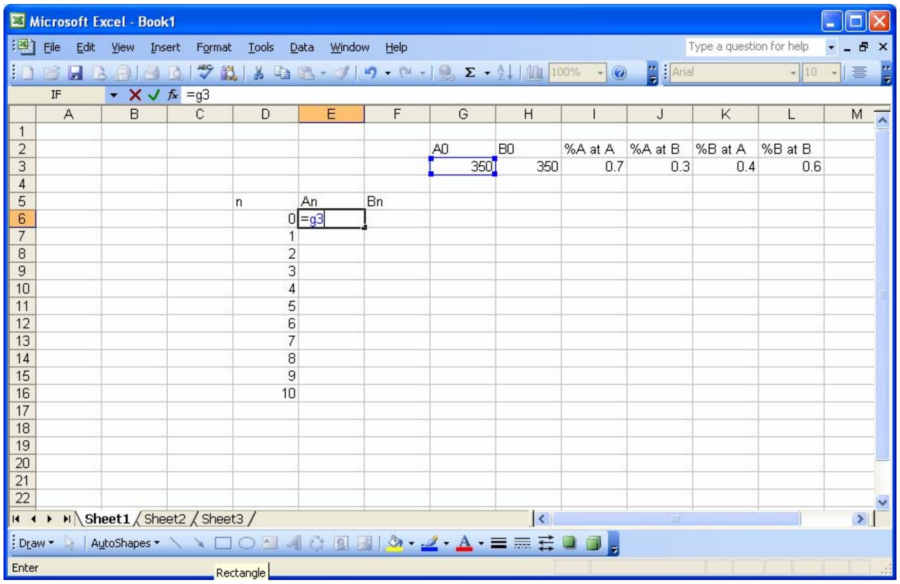
Use whatever headings make sense to you. You're the one writing the formulas, you need to know how to read the sheet. Now put the appropriate values in the appropriate cells.



E6

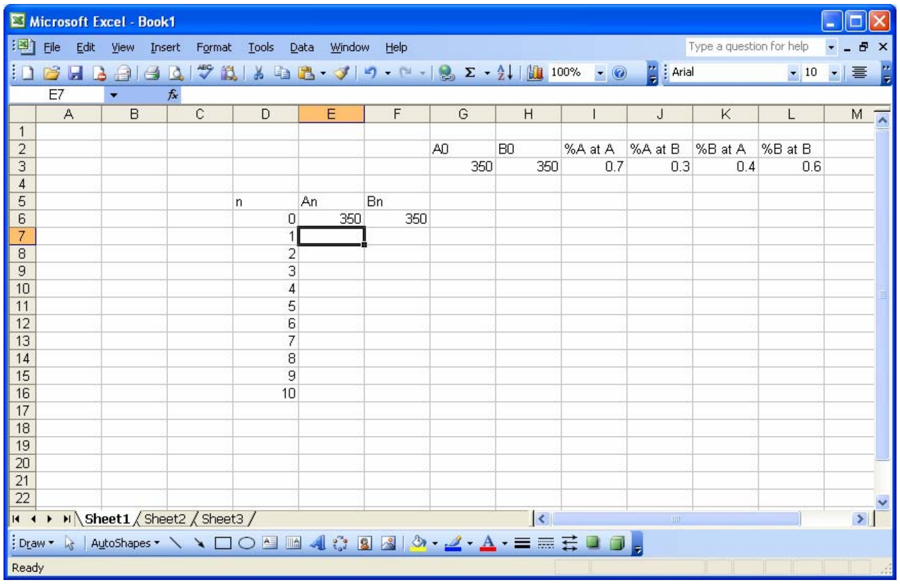
	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2							A0	B0	%A at A	%A at B	%B at A	%B at B	
3							350	350	0.7	0.3	0.4	0.6	
4													
5				n	An	Bn							
6				0									
7				1									
8				2									
9				3									
10				4									
11				5									
12				6									
13				7									
14				8									
15				9									
16				10									
17													
18													
19													
20													
21													
22													

Notice we put the percentages in as decimals. We could have put them in as whole numbers and done the percentages in our calculations; whatever is easier. Now we need to write our formulas. This time we'll use only cell references for our formula. It's a little harder to write this way, but changing starting values will be much easier. Start with the initial values. We've typed them in at cells G3 and H3, but we need them in cells E6 and F6. Let's make that so. To tell Excel we want the value of one cell in another cell, we simply type = and the cell designator where are data is, in this case G3. So select cell E6 and type =*g3*. Like this



	D	E	F	G	H	I	J	K	L
2				A0	B0	%A at A	%A at B	%B at A	%B at B
3				350	350	0.7	0.3	0.4	0.6
5	n	An	Bn						
6	0	=g3							
7	1								
8	2								
9	3								
10	4								
11	5								
12	6								
13	7								
14	8								
15	9								
16	10								

Hit ENTER. The value of cell G3 should now be displayed in cell E6. Change G3 a few times to see if E6 also changes. Now do the same thing for cell F6. We want the value of H3 here, so type =*h3* and hit enter. Your sheet should look like this

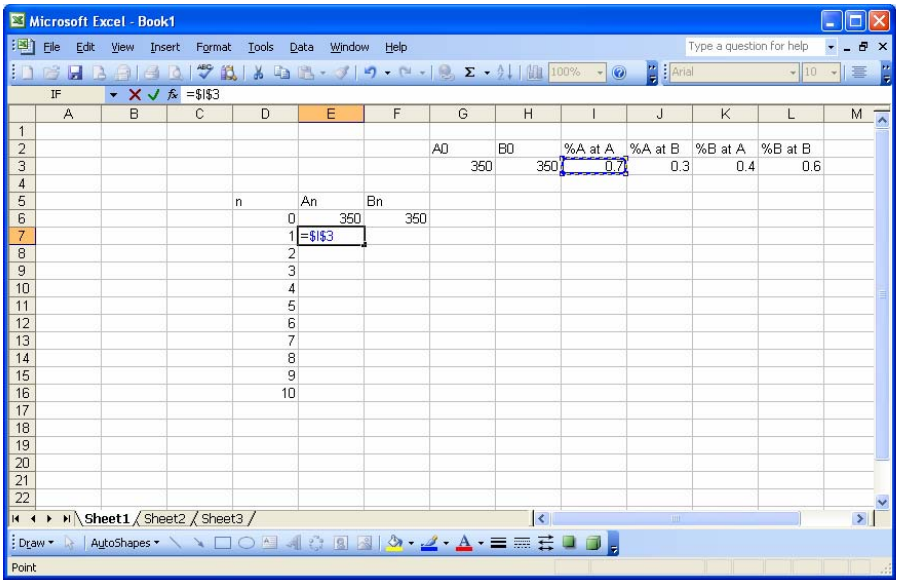


	D	E	F	G	H	I	J	K	L
2				A0	B0	%A at A	%A at B	%B at A	%B at B
3				350	350	0.7	0.3	0.4	0.6
5	n	An	Bn						
6	0	350	350						
7	1								
8	2								
9	3								
10	4								
11	5								
12	6								
13	7								
14	8								
15	9								
16	10								

On to the formula, the first equation is

$$A_{n+1} = 0.70A_n + 0.40B_n$$

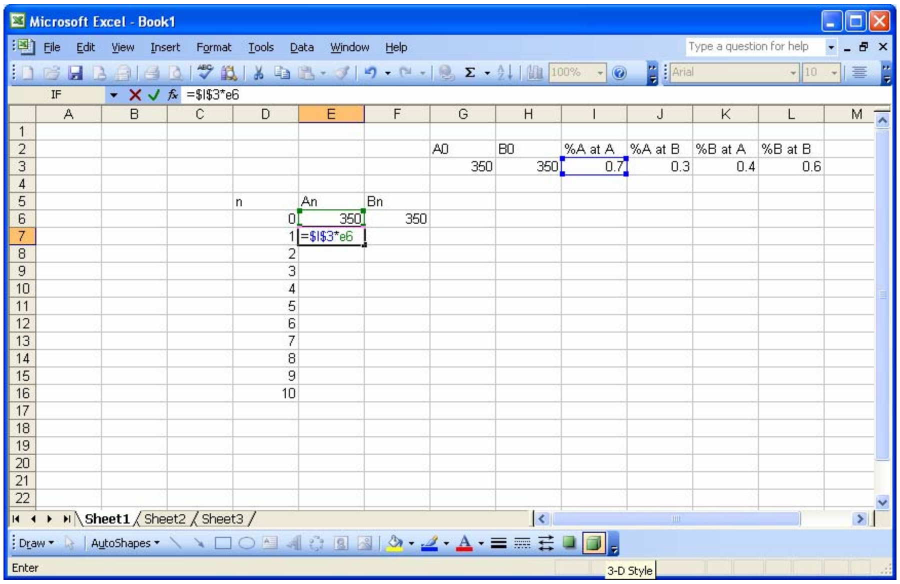
First we need our equals sign to tell Excel we're writing a formula. Next we need to refer to the cell that contains the percentage of vehicles from motor pool A. That is cell I3. Type *i3* or click on that cell. Now, we know that when we are finished with it, we are going to copy this formula, as we did last time, many times in order to work our model out to the end. Last time we copied "relative." In this case though, we don't want to copy relative; we don't want Excel to change the reference point as we copy down our list. We need to let Excel know that. We can do this by pressing the F4 key after typing the cell we want to refer to. When we hit F4, Excel puts dollar signs, $, before the column and row designators. This let's Excel know to maintain that cell reference as it copies. So after typing i3 or selecting it press the F4 key. Your sheet should look like this



IF =$I$3

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2							A0	B0	%A at A	%A at B	%B at A	%B at B	
3							350	350	0.7	0.3	0.4	0.6	
4													
5				n	An	Bn							
6				0	350	350							
7				1	=$I$3								
8				2									
9				3									
10				4									
11				5									
12				6									
13				7									
14				8									
15				9									
16				10									
17													
18													
19													
20													
21													
22													

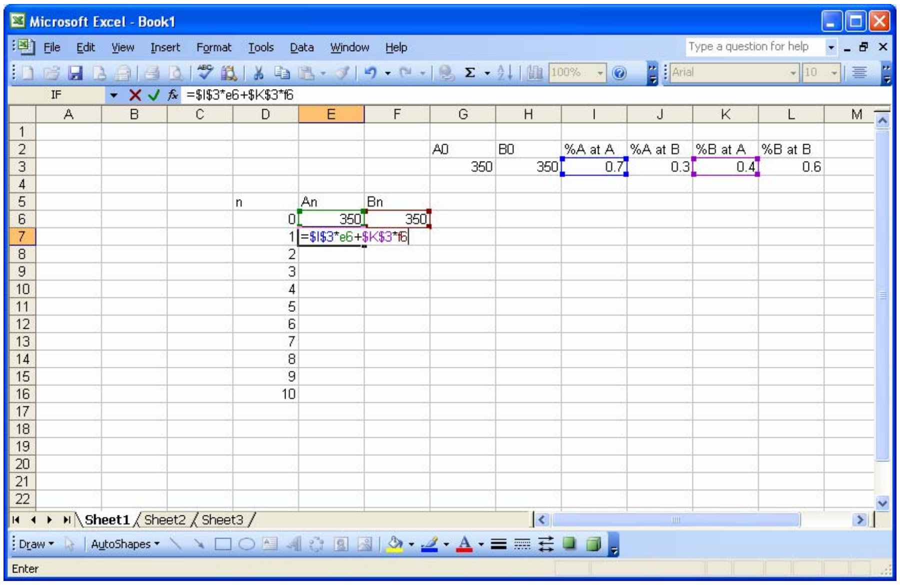
Continuing with our formula, we now want to multiply the percentage by the starting number of vehicles. The starting number is in two places, cell G3 and cell E7, which one should we use? We'll use E7 because we set it to equal G3 earlier. This way we can change our starting values when we are investigating equilibrium values. Type _*e6_. We don't want to press the F4 key this time because when recalculating values we always want to use the latest value of A. Your sheet should look like this



IF =$I$3*e6

	D	E	F	G	H	I	J	K	L
2				A0	B0	%A at A	%A at B	%B at A	%B at B
3				350	350	0.7	0.3	0.4	0.6
5	n	An	Bn						
6	0	350	350						
7	1	=$I$3*e6							
8	2								
9	3								
10	4								
11	5								
12	6								
13	7								
14	8								
15	9								
16	10								

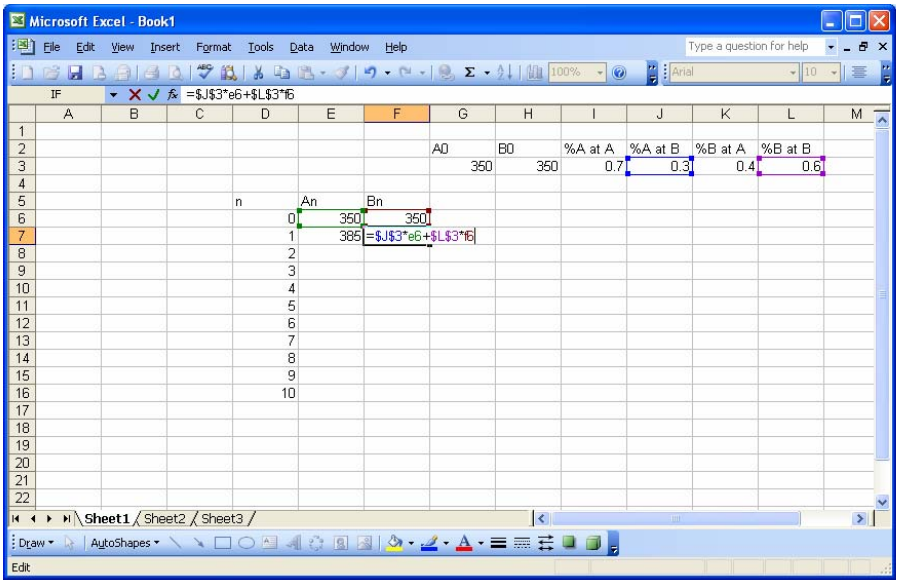
Referring back to our equation, we see that now we need to add the percentage figure for vehicles originally at B. Cell K3 is the percentage and cell F6 holds our original number so we type *+k3*, press the F4 key, type **f6*. Your sheet should look like this



IF =$I$3*e6+$K$3*f6

	D	E	F	G	H	I	J	K	L
2				A0	B0	%A at A	%A at B	%B at A	%B at B
3				350	350	0.7	0.3	0.4	0.6
5	n	An	Bn						
6	0	350	350						
7	1	=$I$3*e6+$K$3*f6							
8	2								
9	3								
10	4								
11	5								
12	6								
13	7								
14	8								
15	9								
16	10								

Press ENTER and we've got our first formula. Take a stab at the formula for *Bn*. When you're done it should look like this

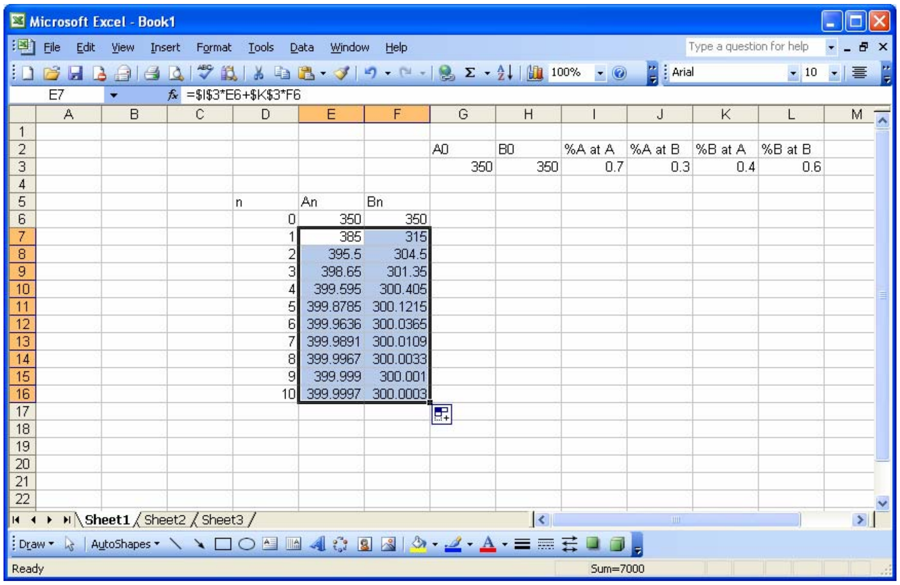


IF =$J$3*e6+$L$3*f6

	D	E	F	G	H	I	J	K	L
2				A0	B0	%A at A	%A at B	%B at A	%B at B
3				350	350	0.7	0.3	0.4	0.6
5	n	An	Bn						
6	0	350	350						
7	1	385	=$J$3*e6+$L$3*f6						
8	2								
9	3								
10	4								
11	5								
12	6								
13	7								
14	8								
15	9								
16	10								

Press ENTER and we're finished with our formulas.

Now we simply need to copy our formulas down far enough to give us that data we need. Select both cells E7 and F7 (click and hold in E7, drag to F7, release). Now click and hold on the square at the bottom right corner of the selected items, drag down and release when ready. If you need more data, you can begin copying where you left off and copy down a few more lines. Your sheet should look like this



E7 =$I$3*E6+$K$3*F6

	D	E	F	G	H	I	J	K	L
2				A0	B0	%A at A	%A at B	%B at A	%B at B
3				350	350	0.7	0.3	0.4	0.6
5	n	An	Bn						
6	0	350	350						
7	1	385	315						
8	2	395.5	304.5						
9	3	398.65	301.35						
10	4	399.595	300.405						
11	5	399.8785	300.1215						
12	6	399.9636	300.0365						
13	7	399.9891	300.0109						
14	8	399.9967	300.0033						
15	9	399.999	300.001						
16	10	399.9997	300.0003						

That's it. We can now easily change our starting values to see how the system responds. Simply type new values under the headings *A0* and *B0*. Try it a few times to see what happens. We could also investigate what would happen to our system if the percentage of vehicles at either motor pool changes (these values, by the way, are called coefficients). Try it, see what happens.

We now have some skills in Excel that will allow us to work some fairly complex models. Go back to some of our other formulas and see if you can figure out how to input them into Excel. You’ll know if you got them right when your data is the same as in the workbook. Next lesson we’ll look at a military application of modeling; the Lanchester model.

VI. LESSON 5 – THE LANCHESTER EQUATIONS OF COMBAT (MODELING ATTRITION)

In lesson 3, we looked at two systems of difference equations that modeled attrition. The first was the Battle of Trafalgar example. The other was the competing insurgency example. Both of these equations showed two forces facing each other, each of which, through one means or another, decreased the size of the other. Now let's take a look at the origin of those equations. F.W. Lanchester investigated air combat situations during World War I. He developed a model of attrition using differential equations to show that the combat casualty rate (attrition rate) for force *X* is proportional to the strength of force *Y* and vice versa. They look like this:

$$\frac{dx}{dt} = -ay, \quad a > 0$$

$$\frac{dy}{dt} = -bx, \quad b > 0$$

Differential equations, thankfully, are beyond the scope of this course, but we've been modeling change using these types of equations all along, except we've been modeling change at discrete time periods while models using differential equations model change continuously. Let's take a quick look at how the equations are related. The left side of Lanchester's equations ($\frac{dx}{dt}$) is modeling the change in *x* (or *y*) with respect to the change in time (*t*). As stated previously, we've been modeling changes over time since we started. For our purposes

$$\frac{dx}{dt} \cong \frac{\Delta x}{\Delta n}$$

where *x* is the value of the system for which we are trying to model change and *n* is the discrete unit of time at which we are modeling the change. (That $\cong$ symbol means approximately equal to.) We know from previous lessons that $\Delta x = x_{n+1} - x_n$ and in our equations so far *n* has changed by one each time, so we can write the following (again for our purposes, the equality here is close enough, so we'll drop the approximately equal to and just call it equal to)

$$\frac{dx}{dt} = \frac{x_{n+1} - x_n}{1} \quad \text{or} \quad \frac{dx}{dt} = x_{n+1} - x_n$$

So, we have an equivalent we can work with to use the Lanchester model for modeling change over discrete time periods. Let's compare:

$$\frac{dx}{dt} = -ay \quad \text{and} \quad \frac{dx}{dt} = x_{n+1} - x_n$$

Since $-ay$ and $x_{n+1} - x_n$ both equal $\frac{dx}{dt}$, we set them equal to each other to get a good approximation of the Lanchester model using a difference equation rather than a differential equation.

$$x_{n+1} - x_n = -ay$$

The constant *a* in Lanchester's model is called the attrition rate coefficient. Now if we solve our equation for the future value of *x*, (x_{n+1}), we get

$$x_{n+1} = x_n - ay$$

Of course, because we are modeling both systems at discrete time intervals, we need a sub *n* on the *y* variable as well, so we have

$$x_{n+1} = x_n - ay_n$$

Since there are two opposing sides in our model, we also need an equation for the attrition of the *Y* force

$$y_{n+1} = y_n - bx_n$$

Again, the constant *b* is an attrition rate coefficient.

These equations, which we derived from the Lanchester model, are exactly the same ones we used to model attrition in the Battle of Trafalgar example in Lesson 3.

$$B_{n+1} = B_n - 0.1F_n$$
$$F_{n+1} = F_n - 0.1B_n$$

To sum up so far, we have some basic equations, derived from the Lanchester model, we can use to model attrition. They are

$$x_{n+1} = x_n - ay_n$$
$$y_{n+1} = y_n - bx_n$$

This a simple model in which we assume that the loss rate of either force is proportional to the number of "firers" on the other force. For example, force *X* may lose 1 person for every 10 firers on force *Y*. Force *Y* may lose 2 people for every 10 firers on force *X* (force *X* has a better aiming system, they're better trained, etc.) The equations would look like this

$$\begin{aligned} x_{n+1} &= x_n - 0.1y_n \\ y_{n+1} &= y_n - 0.2x_n \end{aligned}$$

We can refine our model when we assume an interaction between the forces is necessary for attrition to occur (the firer must locate a target before firing). You'll remember from Lesson 3 that we usually model interactions as the product of the two interactors, so our model would look something like this

$$\begin{aligned} x_{n+1} &= x_n - ax_ny_n \\ y_{n+1} &= y_n - bx_ny_n \end{aligned}$$

(The attrition rates are proportional to the interactions between the two forces.)

The Lanchester model is extremely versatile. You can add practically any variable you need to make your model as accurate as possible. We could further refine our model by adding in some growth (reinforcements, recruits, etc.) We could assume that reinforcements are proportional to the current number of the force. Our model would then look something like this

$$\begin{aligned} x_{n+1} &= x_n + cx_n - ax_ny_n \\ y_{n+1} &= y_n + dy_n - bx_ny_n \end{aligned}$$

The variables *c* and *d* could be called the reinforcement constants. This model is the same as the two competing insurgencies example from Lesson 3

$$\begin{aligned} I_{n+1} &= I_n + 0.15I_n - 0.001I_nS_n \\ S_{n+1} &= S_n + 0.2S_n - 0.002I_nS_n \end{aligned}$$

If you're modeling a tank battle, you could add in a maintenance factor, say 10% of force *X's* tanks are down for maintenance at any one time ($-0.1x_n$). The equations for the different forces don't have to be the same, either. If you're modeling a guerrilla force, *G*, vs. a conventional force, *C*, you may want to show that the guerrillas are only attritted when the two forces interact, but because of the nature of guerrilla warfare, the attrition rate of the conventional force may only be dependent on the number of guerillas, not interaction (in the case of suicide bombers for instance). With reinforcements added in, the formulas may look something like this

$$G_{n+1} = G_n + k_1 G_n - aG_n C_n$$
$$C_{n+1} = C_n + k_2 C_n - bG_n$$

We could go on, but let's just finish up by saying these Lanchester equations can handle just about any attrition model we can think of. We'll make more refinements as we go along in our coursework.

This lesson we took a look at the Lanchester attrition model. We whittled down the differential equation, which models change continuously, to a good approximation of that continuous change modeled at discrete time periods. The Lanchester model will become your model of choice for modeling combat situations. In the next lesson we'll model an arms race to introduce the concept of graphical analysis.

VII. LESSON 6 – GRAPHICAL ANALYSIS

Sometimes it's useful to put your data on a graph to analyze it. The trends of the data show up very well on graphs. Let's look at an arms race between two countries to illustrate.

Suppose you have two countries, Country *X* and Country *Y*, in an arms race. What's that going to look like? To start, each country will have some minimum number of weapons it thinks will deter the other country. Neither country will want to go below that number. Then each country will want to add weapons (this is the arms race part) to ensure that after the other country attacks, it will still retain its original number. In other words each country will make some educated guess about the effectiveness of the other country's weapons and add to its stockpile accordingly.

For example Country *Y* believes it needs 120 weapons to deter the enemy. *Y* also believes that for every 2 weapons *X* has, it needs to add 1 additional weapon. (*Y* believes that it will lose 1 weapon for every 2 weapons *X* fires, so *Y* will need to rebuild according to that ratio.) This will ensure *Y* has at least 120 weapons after any attack. Based on these beliefs about the number of weapons *X* has, the number of weapons *Y* needs (*y* weapons) is

$$y = 120 + \frac{1}{2}x$$

This equation simply says the number weapons *Y* has is equal to 120 (minimum for deterrence) plus 1 for every 2 *X* has (to sustain the minimum.)

Now Country *X* believes it needs 60 weapons to deter the enemy. Also, *X* believes it needs to add 1 weapon for every 3 that *Y* has. Therefore, the mathematical equation for the number of weapons *X* has is

$$x = 60 + \frac{1}{3}y$$

Based on each country's beliefs about the other country's weapons, how does the arms race go?

Let's start at the beginning. Suppose neither country believes the other has weapons and each arms for deterrence. That means at the beginning stage (n = 0) y has

120 weapons and *x* has 60. Assuming each country has perfect intelligence and knows how many weapons the other has, the arms race is on. At *n* = 1 both countries build weapons according to their beliefs about the other country so

$$y = 120 + \frac{1}{2}(60)$$
$$y = 150$$

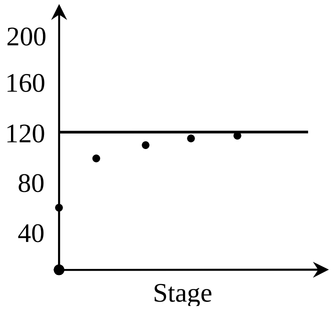
and

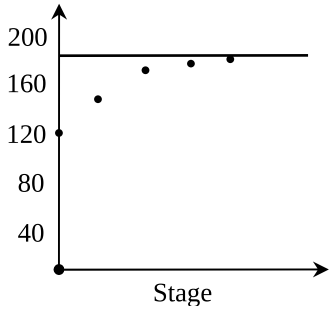
$$x = 60 + \frac{1}{3}(120)$$
$$x = 100$$

The arms race now proceeds in successive stages, each country adding more weapons in response to the other country's increases. Let's take a look at a table of data representing the arms race through 5 stages

n	*y*	*x*
0	120	60
1	150	100
2	170	110
3	175	117
4	178	118

Notice the growth appears to be slowing; *y* appears to be approaching about 180 and *x* appears to be headed to about 120. We can draw up a couple of graphs to show where the data is headed.





These graphs are neatly drawn to make the presentation here clearer, but for what we're doing, we could simply draw a few lines on a piece of paper. We just need to be able to see the trend of the data.

As you can see the graphs give us an “at a glance” sense of what the data is doing. There will be many instances where you’ll want that quick look at the data. In fact, there will be times when this quick look is all you’ll need.

Of course, we’ll also want to be able to discuss the stability of the equilibrium we seem to be experiencing. Is it stable? In other words, will small changes in the initial values have little affect the final outcome? Is it sensitive to changes in the coefficients, in this case the weapons ratios? Unfortunately, graphs won’t quickly tell us the answers to these questions. We’ll need numerical solutions for that. Let’s see what we get.

Again, we need a model to predict future values, $n+1$. We can easily change the equations we have for our arms race into difference equations, like this

$$y = 120 + \frac{1}{2}x$$

and

$$x = 60 + \frac{1}{3}y$$

become

$$y_{n+1} = 120 + \frac{1}{2}x_n$$

and

$$x_{n+1} = 60 + \frac{1}{3}y_n$$

With the initial values of $x_0 = 0$ and $y_0 = 0$ we get a table of data that looks like this: (this, by the way, is an excellent time for you to practice your Excel work)

n	x_n	y_n
0	0	0
1	60	120
2	100	150
3	110	170
4	116.6667	175
5	118.3333	178.3333
6	119.4444	179.1667
7	119.7222	179.7222
8	119.9074	179.8611
9	119.9537	179.9537
10	119.9846	179.9769
11	119.9923	179.9923
12	119.9974	179.9961

It's easy to see that the system approaches the equilibrium we thought it would. But what happens if both countries start off with more than the minimum number of missiles? What if $x_0 = 100$ and $y_0 = 200$? Put these values into your equations (simple to do if you set up Excel right) and see what you get. After you determine if the equilibrium is stable, see if it is sensitive to the coefficients.

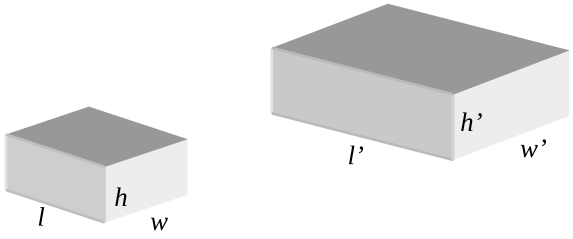
This lesson we looked at the importance of graphing our data. For a quick look at the trends of that data, comparative analysis by graphs can't be beat. We still need to be able to numerically solve our system using difference equations, but the graphs will be useful in many situations. In fact, in a later lesson we'll discuss using these graphs to determine which models "fit" the data. Next lesson we'll examine modeling using geometric similarity.

VIII. LESSON 7 – MODELING USING GEOMETRIC SIMILARITY

Let’s suppose you are put in charge of the local fishing derby. The local club wants to practice catch and release, so to determine the winner you can’t just weigh the fish when they come in and portable scales are not accurate enough. You need a way to predict the weight of a fish using some easily measurable dimension. You need to model using geometric similarity.

Geometric similarity is related to proportionality, which we discussed way back in Lesson One, and is very useful in simplifying the modeling process. In general, objects are geometrically similar if they look alike. More specifically, if there is a one-to-one correspondence between points of two objects so that the ratio of distances between corresponding points is constant for all possible pairs of points, the objects are geometrically similar.

Let’s look at an example. Consider two boxes.



If we assume the boxes are geometrically similar, then we can say that all measures on the boxes are proportional. Therefore l is proportional to *l’*, h is proportional to *h’*, and *w* is proportional to *w’*. Also, they are all related in the same proportion. We can say

$$\frac{l}{l'}=\frac{w}{w'}=\frac{h}{h'}=k$$

where *k* is the constant proportion between all the measures.

Now that we know these measures are proportional, we can see further simplification. Surface area and volume are two areas where the simplification can be seen. Surface area of a solid rectangle is simply the total of the area of each rectangular face. Because there are three sets of equal rectangles; two at $l*h$, two at $w*h$, and two at $l*w$ we can write the formula for surface area as

$$S = 2lh + 2wh + 2lw$$

Therefore, we can write the proportionality as

$$\frac{S}{S'} = \frac{2lh + 2wh + 2lw}{2l'h' + 2w'h' + 2l'w'} = k^2$$

Volume of a solid rectangle is simply the product of the length, the width, and the height. So the formula is

$$V = lwh$$

and the proportionality is

$$\frac{V}{V'} = \frac{lwh}{l'w'h'} = k^3$$

Without going into the math (believe me you don't want to know) we derive the following proportions: surface area is proportional to the square of the length and volume is proportional to the length cubed, like this

$$S \propto l^2 \text{ and } V \propto l^3$$

These proportionalities can be rewritten as the following equations

$$S = kl^2$$
$$V = kl^3$$

These are important, remember them.

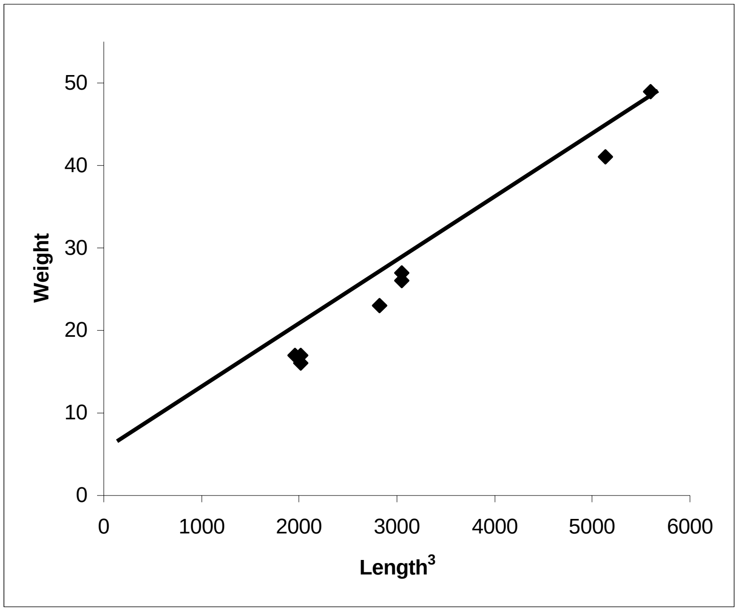
Now back to the fishing derby. There are many assumptions we have to make about the fish we're catching. Neglecting factors like gender and season, we first assume that for each species of fish the average weight density is constant. In other words, the stuff that makes up the weight of a largemouth bass (skin, bone, flesh) is the same from one largemouth bass to another. The next assumption we have to make is that all largemouth bass are geometrically similar. We don't care what shape, but that these fish are simply scale models of one another. These assumption allows us to predict weight based on size (volume) using the equation $V = kl^3$. Now weight is volume times density and were are assuming constant density, so we can substitute weight for volume and use this equation

$$W = kl^3$$

All we need to do is determine the constant. We'll do that by plotting some observed data and determining the slope of the plotted line as we did in Lesson Two. The slope will give us the proportionality and therefore the constant. Here's the observed data: (You spent a nice day on the lake!)

Length, l(in.)	14.5	12.5	17.25	14.5	12.625	17.75	14.125	12.625
Weight, W(oz.)	27	17	41	26	17	49	23	16

Here's what the line looks like:



All proportionalities go through the origin, (0,0), so we're good there. We have a point at approximately (3000,25) so using the formula for the slope, we get

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$m = \frac{25 - 0}{3000 - 0}$$

$$m = \frac{25}{3000}$$

$$m = 0.00833$$

This means our constant is estimated to be 0.00833 and yields the following model

$$W = 0.00833l^3$$

Now you have enough information to develop a simple chart of lengths vs. weights. Your derby entrants can now measure the length of their fish, estimate the weights of their fish, throw the fish back, and still have a qualifying weight for the derby. The chart can be developed by multiplying some random lengths by the constant to determine the corresponding weights. Your chart might look like this

Length, l(in.)	12	13	14	15	16	17	18	19	20	21	22	23
Weight, W(oz.)	14	18	23	28	34	41	49	57	67	77	89	101
Weight, (lb)	0.9	1.1	1.4	1.8	2.1	2.6	3.0	3.6	4.2	4.8	5.5	6.3

We've done it. We've given our derby entrants a relatively simple way to figure out how much their fish weigh and we've satisfied the conservationists who wanted us to throw the fish back. Now if we can keep PETA off our backs for catching the fish in the first place. Fish have feelings too, you know!

Let's do one more. PETA won't be too happy about this one either! Suppose you have a 20 pound pink flamingo. It stands 3 feet high and has legs that measure 2 feet. Let's model the height and leg length of a 100 pound flamingo.

First we have to make a few assumptions: 1) All flamingos have the same shape and therefore a relative volume, 2) All flamingos have an equal density, i.e., same stuff makes up body mass, 3) the ratio of legs height to total height is the same for all flamingos. These seem reasonable.

We already know a formula for volume using the length measurement, it is

$$V = kl^3$$

Because density is constant, weight and volume are equal so

$$W = kl^3$$

We'll use the data from the measured flamingo to calculate our constant. Simply plug in the numbers

$$
\begin{aligned}
20 &= k3^3 \\
20 &= k27 \\
\frac{20}{27} &= k \\
0.74 &= k
\end{aligned}
$$

Now plug *k* into the equation for the 100 pound flamingo

$$
\begin{aligned}
W &= kl^3 \\
100 &= 0.74l^3 \\
\frac{100}{0.74} &= l^3 \\
\sqrt[3]{135.14} &= \sqrt[3]{l^3} \\
5.13 &= l
\end{aligned}
$$

So, we have a flamingo that is 5.13 feet tall. We can calculate the leg height using the ratio of total height to leg height from the first flamingo. We simply compare the ratios and solve for the unknown.

$$
\begin{aligned}
\frac{3}{2} &= \frac{5.13}{x} \\
\frac{3}{2}x &= 5.13 \\
x &= \frac{5.13*2}{3} \\
x &= 3.42
\end{aligned}
$$

We end up with a 100 pound flamingo that is 5.13 feet tall whose legs are 3.42 feet long.

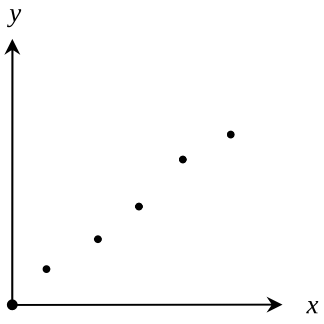
Alright, that’s geometric similarity in a nutshell. Once again, it is extremely useful in simplifying the modeling process. Instead of trying to calculate the volume of oddly shaped objects in order to compare them, we can simply, if the objects are similar, use one measure of the object to do the comparison. Much easier, much faster. Next lesson we’ll examine model fitting.

THIS PAGE INTENTIONALLY LEFT BLANK

IX. LESSON 8 – MODEL FITTING

So far, we have examined modeling from a numeric standpoint. We were given some data and we determined a mathematical model from that data. In this lesson, we're going to fit a model to the data as presented graphically. There are a couple of "rules of thumb" we'll use to determine what mathematical model best fits our graphed data. We'll look at data that, when graphed, forms a line and data that forms a curve. We'll also look at one method for determining the "best fit" line. The main idea here is to make you familiar with the way certain equations look on a graph, so when you see the graph of some data, you'll be able to predict what model will best fit it. Let's get started.

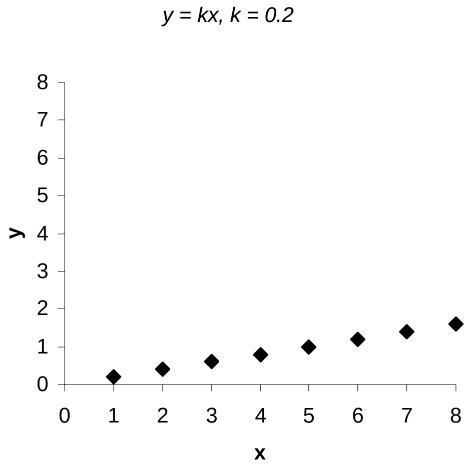
As we learned in Lesson Six, graphs are an excellent tool for seeing the general trend of data. By looking at graphed data we can quickly see what the data is "doing." From this observation of graphed data we can also begin to determine which mathematical model best represents our data. Let's take look at some graphed data to see how.

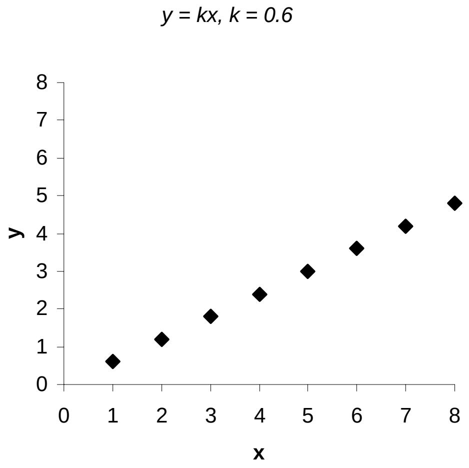


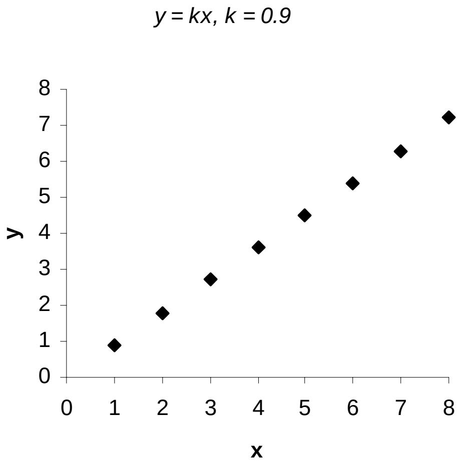
These data points seem to lie on a fairly straight line. But two related problems present themselves. First, what model can we use to represent this data? Then, where is the "best fit" line? Let's start with the first question. Since we are proposing that this is indeed a line (as opposed to a curve), we should use a linear equation. The line on the graph above appears to go through the origin, (0,0), of our graph. This fact suggests using a proportional model since all proportions make lines that go through the origin. Remember, two variables are proportional when one is always some constant multiple of the other. The general equation for a proportion is:

$$y = kx$$

Here's an example of some graphs for this proportion. We've varied *k* so we can see the difference the value of the constant makes. Remember, *k* is the slope.

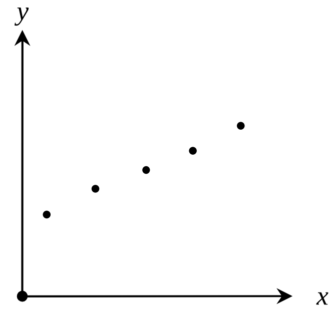






So, if you've got a line that goes through the origin, your best bet for a model is a simple proportion. We used a proportional model like this when we modeled changes in population growth of a herd of animals way back in Lesson Two. We'll look at proportions with exponents like $y = kx^2$ a little later.

What if the data appears to be a line, but doesn't seem to go through the origin, like this?



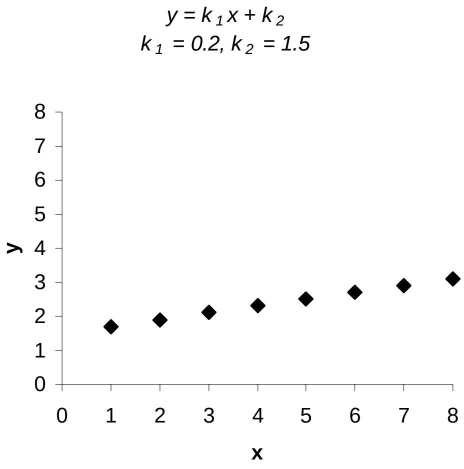
A proportion would not be appropriate because the line does not go through the origin. We should use an equation that looks similar but has an added element.

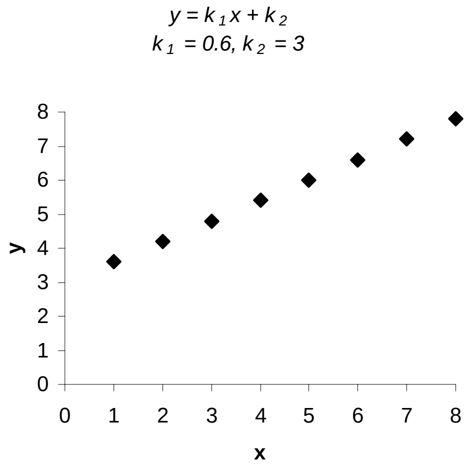
$$y = k_1 x + k_2$$

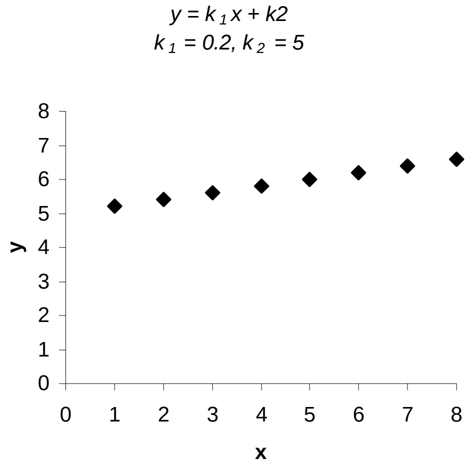
This equation may look familiar to you. If we substitute different variables for our constants, you'll see it is the same as the equation for a line

$$y = mx + b$$

You might remember from several of our lessons that *m* is the slope. In a few of our examples, we calculated *m* and used it as our constant, *k*. The first part of the equation is a simple proportion, but we've got another variable. In this equation *b* represents the "*y* intercept" or the value on the *y* axis where the line crosses it or put yet another way the *y* coordinate when the *x* coordinate is 0. You can see from the graph above that when *x* is 0, *y* is something greater than 0. We therefore have to add something to *x* to give us the value of the *y* coordinate. That something is *b*. Let's take a look at some graphs for this equation. We'll again vary our constants (k_1 and k_2) so we can see the difference changes will make.



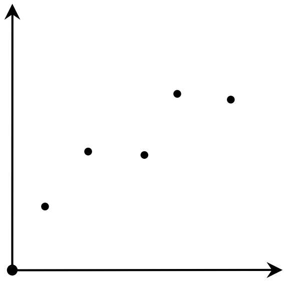




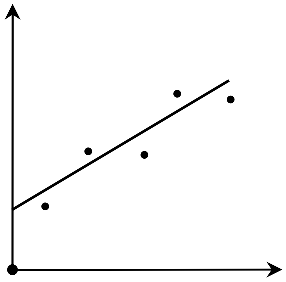
There. If you've got a line, but it doesn't go through the origin, you'll simply use the equation for a line taking into account the *y* intercept. We used an equation like this in modeling unrestricted population growth of our herd in Lesson Two.

Alright, let's address the second problem presented earlier. You've got your data points and you think some sort of line would be the best for a model. But where should you put the line? Its location will play a big part in predicting future values from your model; it needs to be right. Let's take a look at an example.

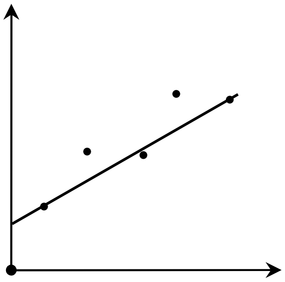
Suppose you've got some data on a graph, like this.



It's fairly clear that we need a line, but where should it be? Here?

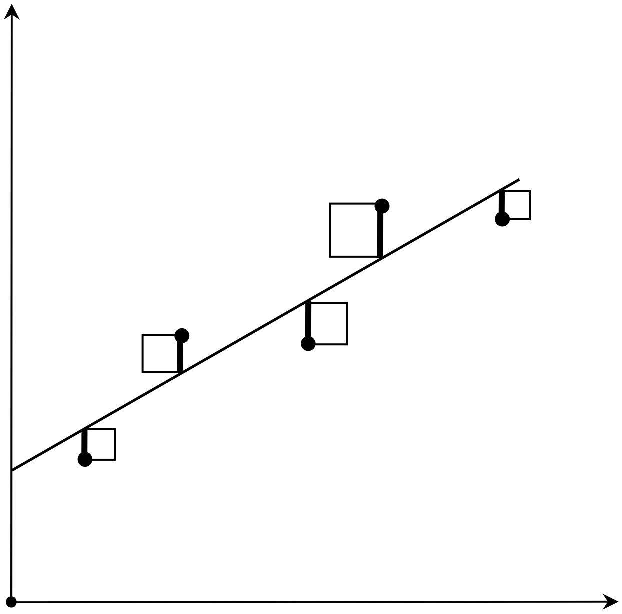


This line doesn't include any of the points, but it is somewhat centered. Would here be better?



This line at least includes two of the points. Those other two points seem to be way off, though.

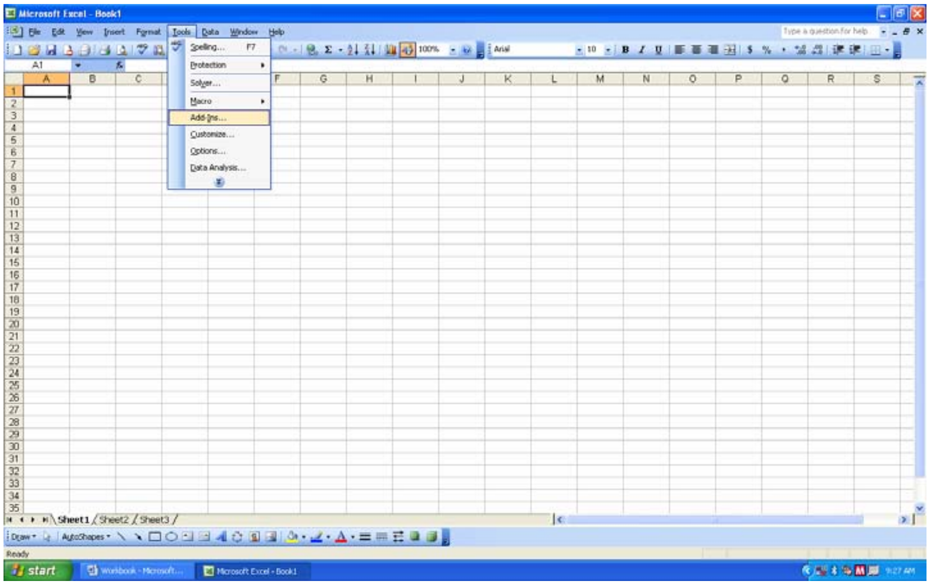
Fortunately, there is a simple method to determine where the best line should be. We'll call it the sum of least squares method. Since you have done a "best guess" estimate on where the line should go, we'll simply find the difference between each of your actual *y* coordinates and the *y* coordinates of your estimated line (the vertical difference as denoted by the heavy lines below), square the difference, and add up the area of the squares formed. Like this



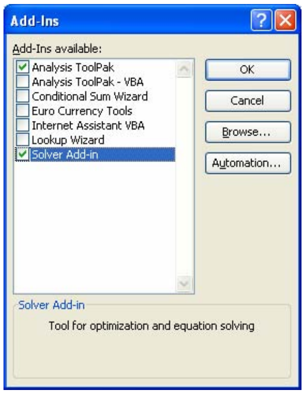
Then you find a line that minimizes the sum of all the squared differences. How do you do that? You could draw a line, calculate; move the line, calculate; move the line, calculate; and continue that until you find the best line, but it would take you a while. Better to let Excel do it for you.

Before we begin, we need to install an Excel add-in called Solver. Solver does exactly what it says it does: it solves problems for you. Basically, you tell Solver what cell has the answer you're trying to get (the target) and whether to maximize the value of that cell, minimize it or have it equal a certain number. Then you tell Solver a range of cells it can change in order to achieve the value you want for your target cell. Solver then runs the problem through a bunch of times until it gets the value you asked for. This sounds easy and it is, but you need to give Solver something to work with; you need to have a pretty close estimate of the target cell value to start. Solver won't work miracles for you. Anyway, let's install it.

Open Excel, click on Tools in the Toolbar at the top of the page. Select Add-Ins…



Excel presents you with a list of add-ins to install. Select Solver Add-in and click OK.



Add-Ins available:

☑ Analysis ToolPak
☐ Analysis ToolPak - VBA
☐ Conditional Sum Wizard
☐ Euro Currency Tools
☐ Internet Assistant VBA
☐ Lookup Wizard
☑ Solver Add-in

Solver Add-in

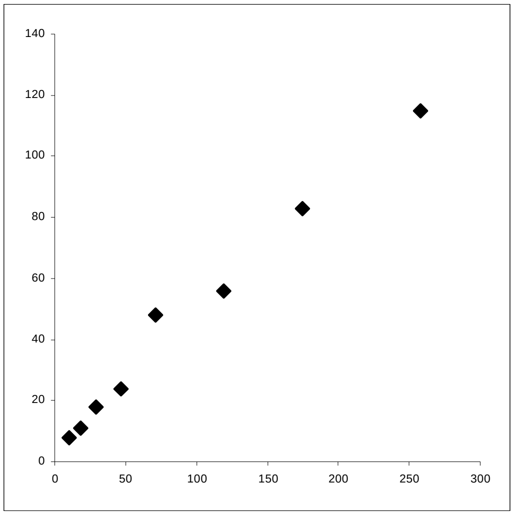
Tool for optimization and equation solving

Excel installs Solver and adds it to the Tools menu. Now you're ready to solve.

You've have some data on a graph. Let's use the data from our population example in Lesson Two. Here it is.

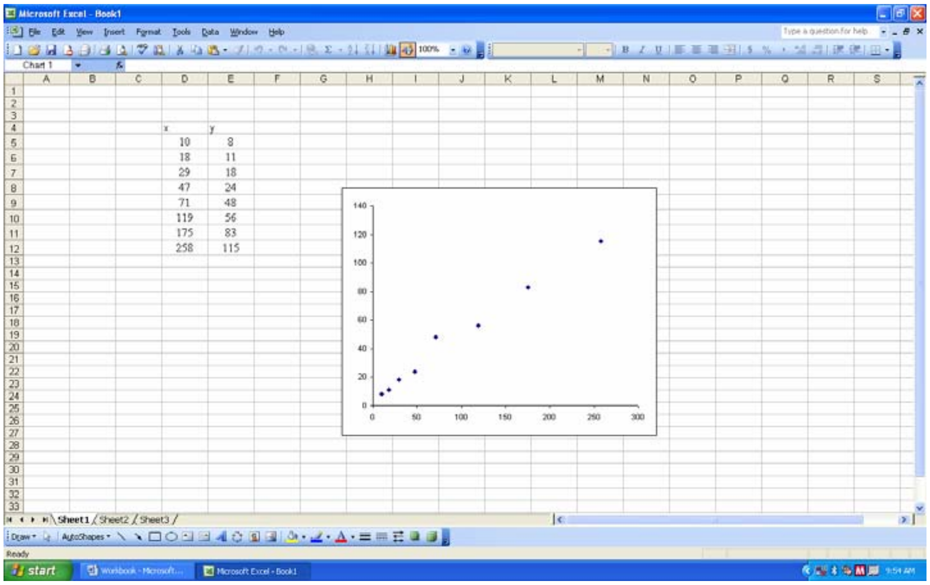
Observed Population	Change in Population
10	8
18	11
29	18
47	24
71	48
119	56
175	83
258	115

Open up Excel and enter this data. Use *x* and *y* as the column headings. Build a scatterplot from the data using the Chart Wizard. It should look something like this.



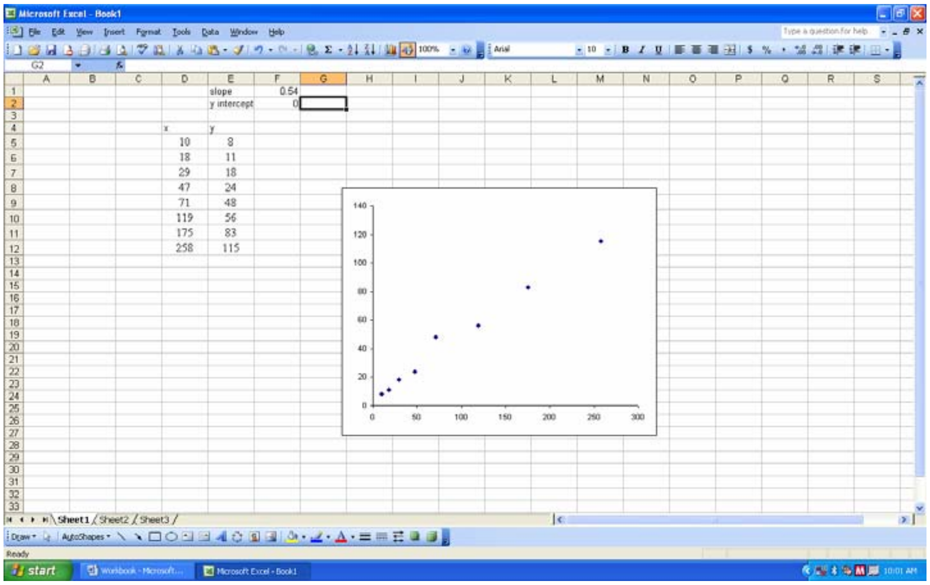
The data appears to be a line, but we can’t visually determine a best fit line. Let’s have Solver do it for us.

So far, your spreadsheet should look something like this.



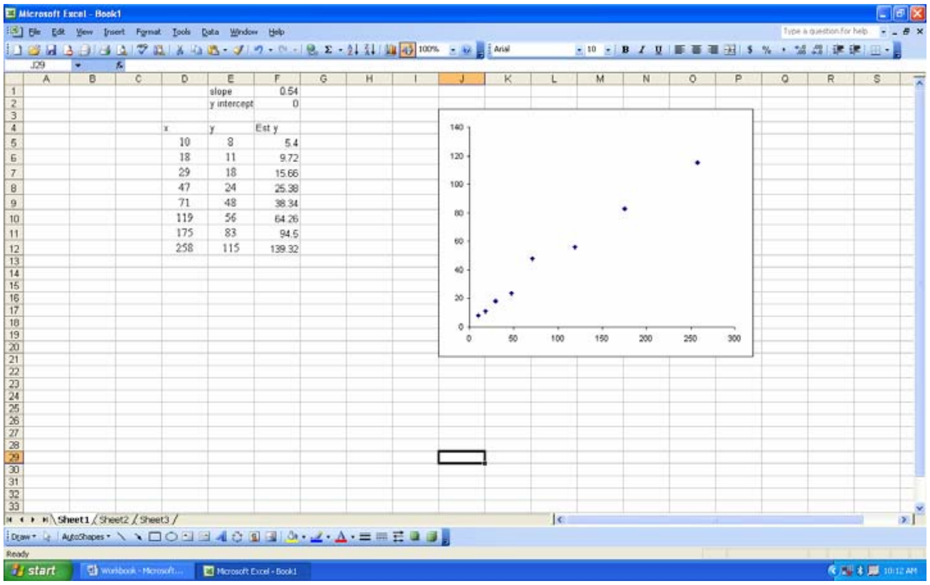
x	y
10	8
18	11
29	18
47	24
71	48
119	56
175	83
258	115

Now we’re going to ask Solver to minimize our squared differences (of the *y* coordinates) by changing cells that contain our slope and our *y* intercept. You’ll remember that the *y* intercept is the *y* coordinate where the line crosses the *y* axis. So what slope and *y* intercept do we use? Well, we just need a rough estimate to get Solver started, so estimate both values based on your graph (don’t forget the formula for slope: $m = \frac{y_2 - y_1}{x_2 - x_1}$) and put these two values in the spreadsheet (somewhere convenient). In Lesson Two, we used 0.54 as an estimated slope for this line, let’s use that. It also seems from looking at the graph, that the line will intercept the *y* axis at the origin, so let’s use 0 for the *y* intercept. Add these to your spreadsheet now. It should look like this



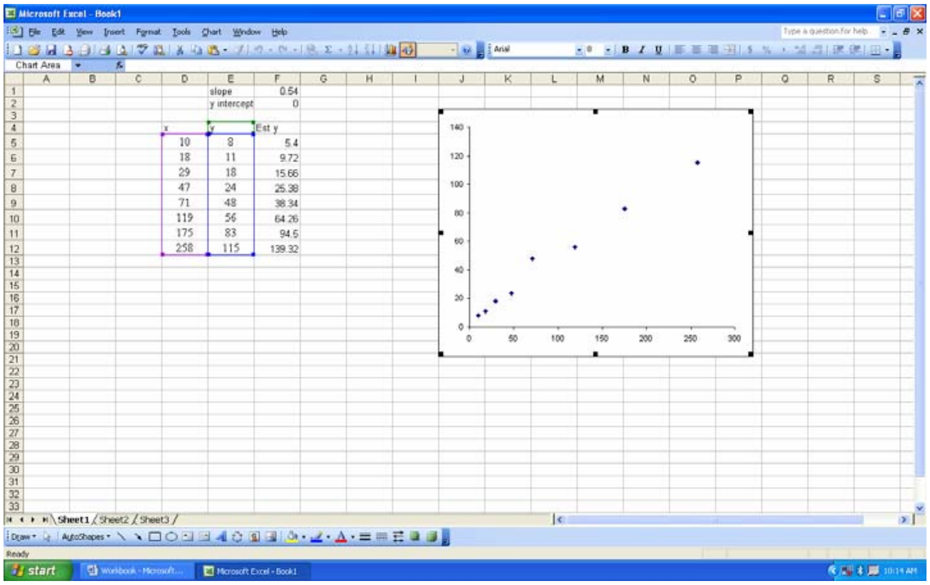
	D	E	F
1		slope	0.54
2		y intercept	0
3			
4	x	y	
5	10	8	
6	18	11	
7	29	18	
8	47	24	
9	71	48	
10	119	56	
11	175	83	
12	258	115	

Now we want Excel to calculate the difference between our actual *y* coordinates and the *y* coordinates of our estimated line. We need some estimated *y* coordinates. We will calculate these by plugging the actual *x* coordinates and our estimated slope and *y* intercept into the equation for a line. This will give us a line based on our estimated slope. The equation is $y = mx + b$. We'll use our estimated slope for *m*, the actual *x* coordinate, and our estimated *y* intercept for *b*. We'll write a formula in Excel in cell F5, and then copy it down for the rest of our data. The formula should look like this: *=$F$1*D5+$F$2*. Your cell references might be different based on what cells you used to hold the slope and *y* intercept values. Once you have the formula right, copy it down. Your spreadsheet should look like this.



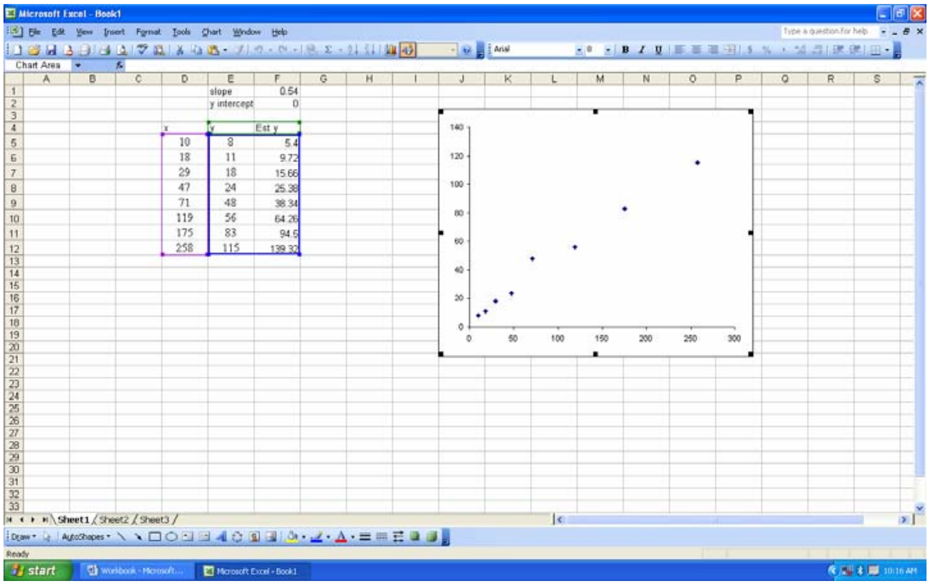
D	E	F
	slope	0.54
	y intercept	0
x	y	Est y
10	8	5.4
18	11	9.72
29	18	15.66
47	24	25.38
71	48	38.34
119	56	64.26
175	83	94.5
258	115	139.32

If you'd like, you can add the estimated line to your graph. This let's you quickly see how close your estimated line is to the actual data. Click on the graph. Excel will highlight it and put borders around the data that makes up the graph. Like this



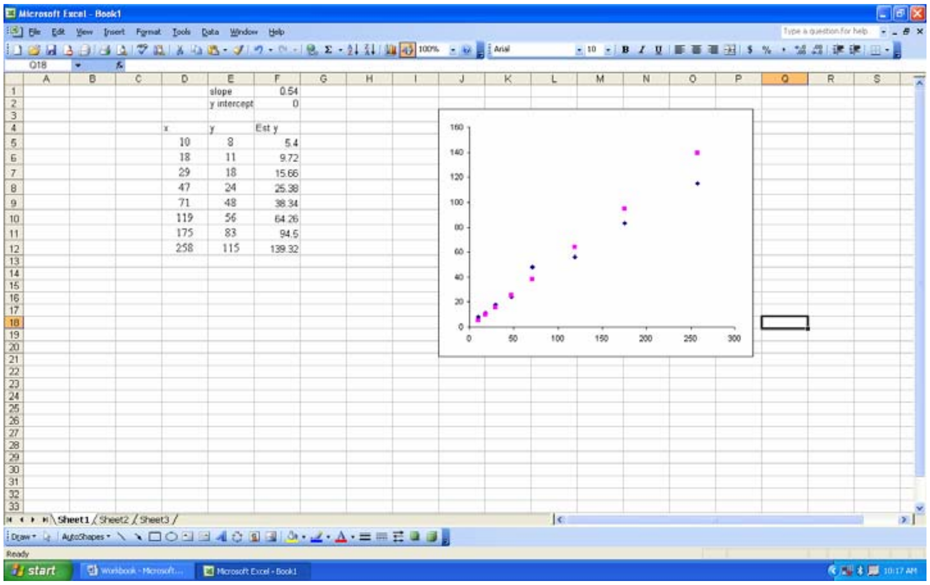
D	E	F
	slope	0.54
	y intercept	0
x	y	Est y
10	8	5.4
18	11	9.72
29	18	15.66
47	24	25.38
71	48	38.34
119	56	64.26
175	83	94.5
258	115	139.32

To add your new *Est y* column of data to your graph, simply click on the corner of the blue border around your data and drag the border around the new data, like this



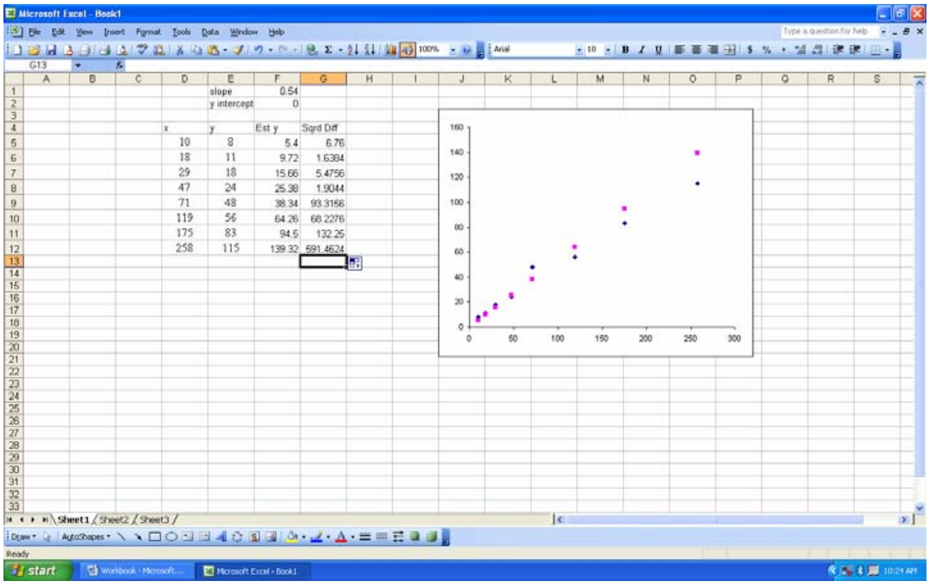
	D	E	F
1		slope	0.54
2		y intercept	0
3			
4	x	y	Est y
5	10	8	5.4
6	18	11	9.72
7	29	18	15.66
8	47	24	25.38
9	71	48	38.34
10	119	56	64.26
11	175	83	94.5
12	258	115	139.32

When you release the mouse, the new data will be included on your graph. Like this



	D	E	F
1		slope	0.54
2		y intercept	0
3			
4	x	y	Est y
5	10	8	5.4
6	18	11	9.72
7	29	18	15.66
8	47	24	25.38
9	71	48	38.34
10	119	56	64.26
11	175	83	94.5
12	258	115	139.32

Now we need to add a column for our squared differences. We'll simply write a formula to subtract the actual *y* from the estimated *y* and square the difference. The formula will look like this *=(F5-E5)^2*. Again, your cell references may be different. We used parentheses to ensure the correct order for calculations. The ^ symbol is used to raise a value to a power. The power is the number after the ^. To square a value use ^2, to cube it use ^3. Once you have the formula right, copy it down. Your spreadsheet should look like this

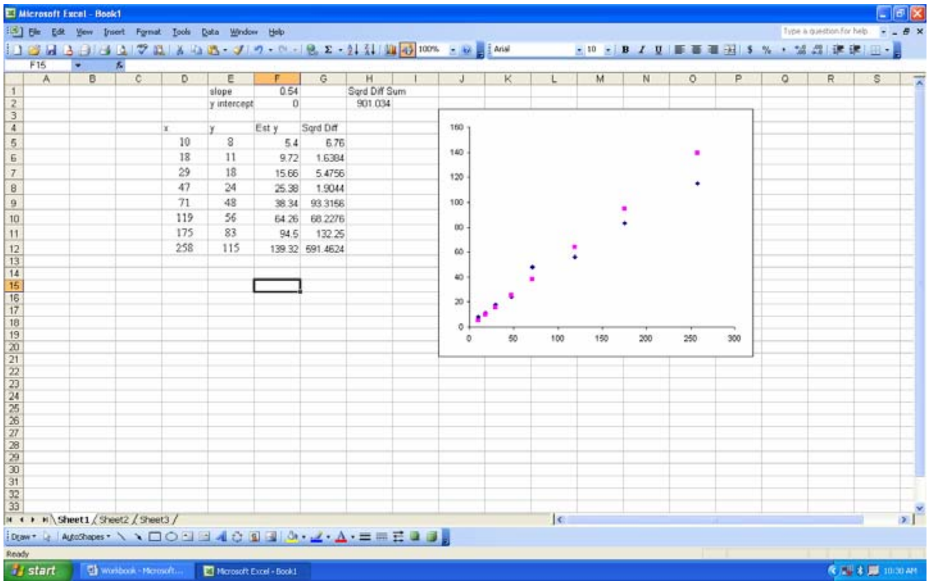


Microsoft Excel - Book1

G13

	D	E	F	G
1		slope	0.54	
2		y intercept	0	
3				
4	x	y	Est y	Sqrd Diff
5	10	8	5.4	6.76
6	18	11	9.72	1.6384
7	29	18	15.66	5.4756
8	47	24	25.38	1.9044
9	71	48	38.34	93.3156
10	119	56	64.26	68.2276
11	175	83	94.5	132.25
12	258	115	139.32	591.4624

Don’t worry about the size of your error, Solver will minimize it for you shortly. Next, we need to sum the squared differences. We’ll put the sum up near the top so we can easily work with it later. Tell Excel you want to sum and then give it a range to sum. The formula should look like this, *=SUM(G5:G12)*. After adding the sum of the squared differences to the spreadsheet, it should look like this



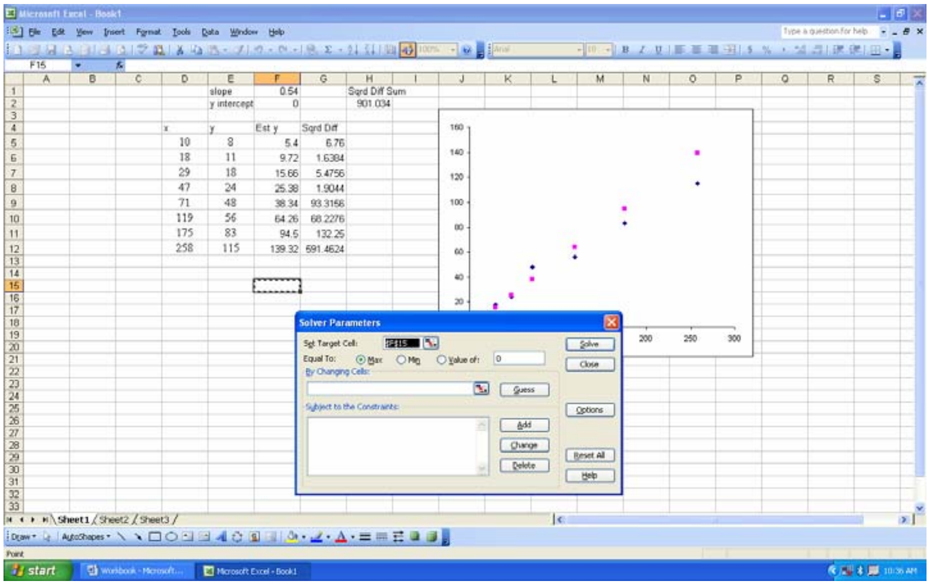
Microsoft Excel - Book1

F15

	D	E	F	G	H
1		slope	0.54		Sqrd Diff Sum
2		y intercept	0		901.034
3					
4	x	y	Est y	Sqrd Diff	
5	10	8	5.4	6.76	
6	18	11	9.72	1.6384	
7	29	18	15.66	5.4756	
8	47	24	25.38	1.9044	
9	71	48	38.34	93.3156	
10	119	56	64.26	68.2276	
11	175	83	94.5	132.25	
12	258	115	139.32	591.4624	

Okay, we’re ready to invoke Solver. We’re going to ask it to minimize cell H2 (the sum of squared differences) by changing cells F1 and F2 (the estimated slope and *y* intercept). Remember, we have a line based on an estimated slope and *y* intercept. Solver is going to find us a better (the best) line by using a new slope and *y* intercept to give us less (the least) error. Here we go.

Click on Tools from the toolbar at the top of the page and select Solver… . A dialog box will open.



	D	E	F	G	H
1		slope	0.54		Sqrd Diff Sum
2		y intercept	0		901.034
3					
4	x	y	Est y	Sqrd Diff	
5	10	8	5.4	6.76	
6	18	11	9.72	1.6384	
7	29	18	15.66	5.4756	
8	47	24	25.38	1.9044	
9	71	48	38.34	93.3156	
10	119	56	64.26	68.2276	
11	175	83	94.5	132.25	
12	258	115	139.32	591.4624	

Solver Parameters

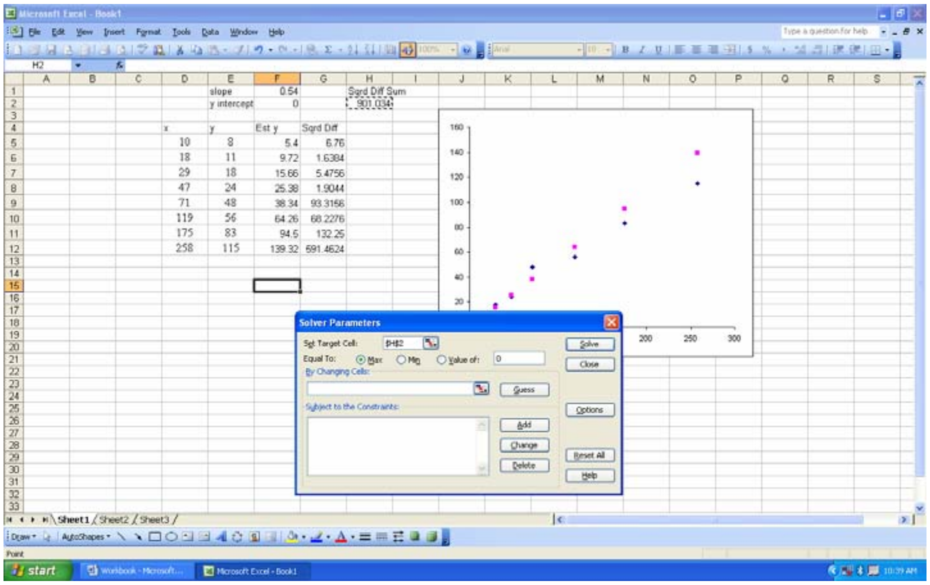
Set Target Cell: $F$15

Equal To: ● Max ○ Min ○ Value of: 0

By Changing Cells:

Subject to the Constraints:

Notice that the cell you had selected when you invoked solver is highlighted on the spreadsheet and its designator is already put into the Set Target Cell: window for you. Also note that the Set Target Cell: window is highlighted, ready for you to enter a new cell. Select the cell you want Solver to target. In our case cell H2, like this



	D	E	F	G	H
1		slope	0.54		Sqrd Diff Sum
2		y intercept	0		901.034
3					
4	x	y	Est y	Sqrd Diff	
5	10	8	5.4	6.76	
6	18	11	9.72	1.6384	
7	29	18	15.66	5.4756	
8	47	24	25.38	1.9044	
9	71	48	38.34	93.3156	
10	119	56	64.26	68.2276	
11	175	83	94.5	132.25	
12	258	115	139.32	591.4624	

Solver Parameters

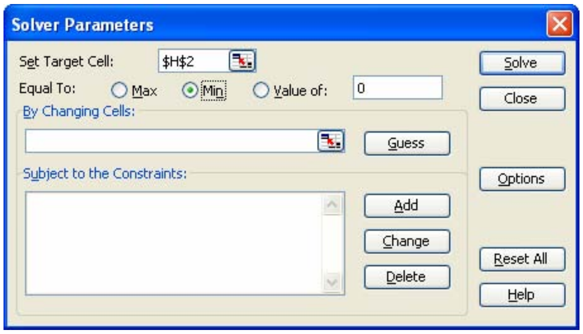
Set Target Cell: $H$2

Equal To: ● Max ○ Min ○ Value of: 0

By Changing Cells:

Subject to the Constraints:

Next, tell Solver what to do with that target cell by selecting an option in the Equal To: row. We want to select Min because we want the least sum of squared differences. There may be other instances when you want solver to maximize the targeted value or you want the targeted cell to equal a particular value. Select the appropriate option and enter a value if necessary.



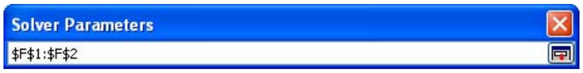
Solver Parameters

Set Target Cell: $H$2

Equal To: ○ Max ● Min ○ Value of: 0

By Changing Cells:

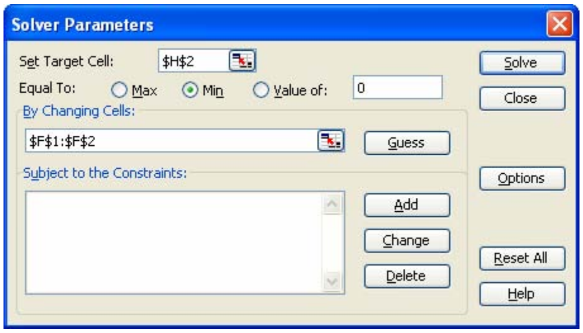
Now we need to enter the cells we want Solver to change in order to minimize the sum of squared differences. In this case, we want to compute a new line that minimizes the sum of squared differences, so we want Solver to change the slope and *y* intercept to give us a line that will minimize the differences between its *y* coordinates and the *y* coordinates of our data. Type the cell designators into the By Changing Cells: window. If you want Solver to work with a range of cells, type the designator for the first cell in the range, type a colon (:), then type the designator for the last cell in the range. You can quickly put cell designators in by clicking in the By Changing Cells: window and selecting the cells you want. As you're selecting cells, Solver will give you a window to show you what cells you're selecting.



Solver Parameters

$F$1:$F$2

When you're finished your Solver window should look like this



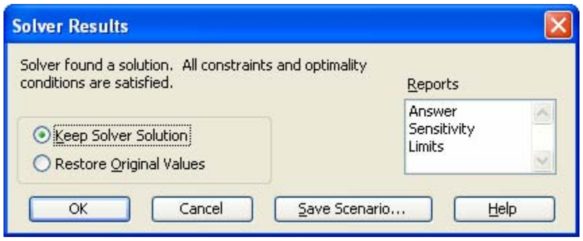
Solver Parameters

Set Target Cell: $H$2

Equal To: ○ Max ● Min ○ Value of: 0

By Changing Cells: $F$1:$F$2

You're ready. Click Solve and Solver goes to work. After crunching the possibilities, Solver gives you a proposed solution and asks if you want to keep the solver solution.



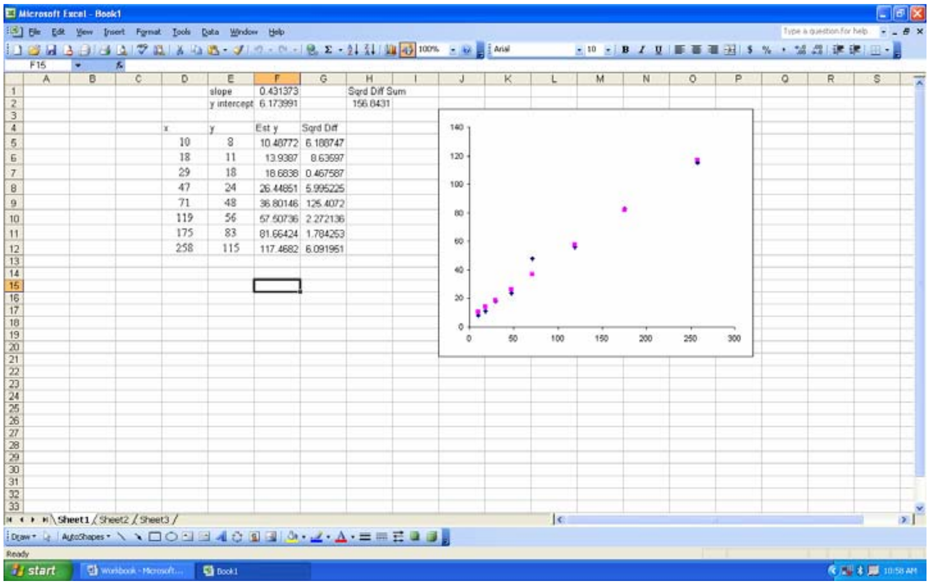
Solver Results

Solver found a solution. All constraints and optimality conditions are satisfied.

● Keep Solver Solution
○ Restore Original Values

Take a look at the solution Solver gave you. In most cases, it'll be fine. Sometimes, however, it'll be way out of whack. You may need to go back and give Solver a closer estimate to start with if this happens. Also, you may have given Solver incorrect parameters to work with. Remember, trash in, trash out. Generally, you can select Keep Solver Solution and click OK.

Your spreadsheet should look like this

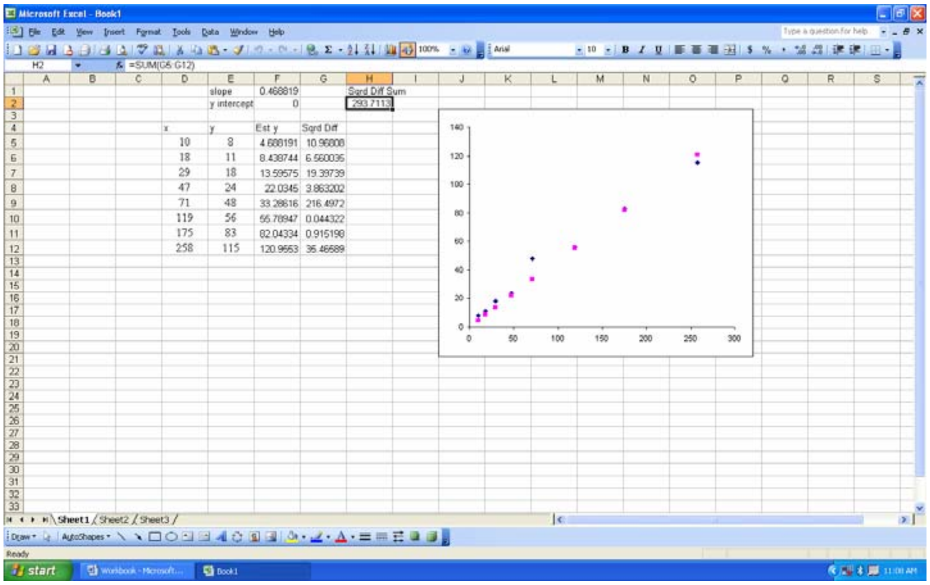


Microsoft Excel - Book1

	D	E	F	G	H
1		slope	0.431373		Sqrd Diff Sum
2		y intercept	6.173991		156.8431
3					
4	x	y	Est y	Sqrd Diff	
5	10	8	10.48772	6.188747	
6	18	11	13.9387	8.63597	
7	29	18	18.6838	0.467587	
8	47	24	26.44851	5.995225	
9	71	48	36.80146	125.4072	
10	119	56	57.50736	2.272136	
11	175	83	81.66424	1.784253	
12	258	115	117.4682	6.091951	

Thanks to Excel and your great estimating abilities, you now have the "best fit" line for your model. Using the slope and *y* intercept Solver calculated for us, the resulting model would be $y = 0.43x + 6.17$. You can now accurately predict *y* for any value of *x*.

Let's look at one more aspect before we move on. Our original data was based on a proportion; $y = kx$. Proportions always give a line that has a *y* intercept of 0. For illustration purposes, we allowed Solver to change the slope and the *y* intercept. Since we know this is a proportion, we should keep the *y* intercept at 0. Like this



H2 =SUM(G5:G12)

	D	E	F	G	H
1		slope	0.468819		Sqrd Diff Sum
2		y intercept	0		293.7113
3					
4	x	y	Est y	Sqrd Diff	
5	10	8	4.688191	10.96808	
6	18	11	8.438744	6.560036	
7	29	18	13.59575	19.39739	
8	47	24	22.0345	3.863202	
9	71	48	33.28616	216.4972	
10	119	56	55.78947	0.044322	
11	175	83	82.04334	0.915198	
12	258	115	120.9553	35.46589	

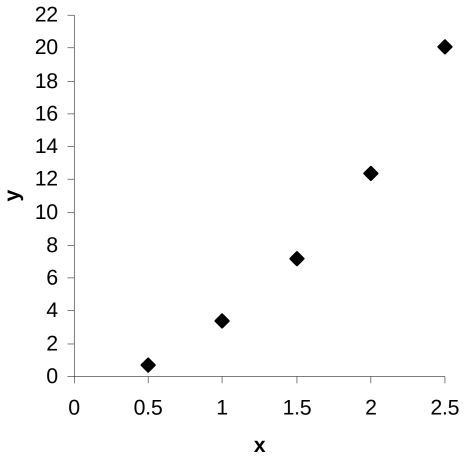
We used the same process, except we set the y intercept to 0 and only allowed Solver to change the slope, cell F1. Note that although the sum of squared differences is larger, this model for our known proportion is more accurate because the line goes through the origin. We'll come back to Solver a lot in the future.

So, we've looked at models for data that graphs out to a line that has a y intercept of 0, $y = kx$; we've also looked at models for data graphing out to a line with a y intercept at something other than 0, $y = k_1 x + k_2$; and we've shown how to find the "best fit" for these lines, but what about data that graphs to a curve? What models are best for those? Let's take a look.

Suppose this is your data

x	y
0.5	0.7
1	3.4
1.5	7.2
2	12.4
2.5	20.1

When plotted, it yields a curve like this

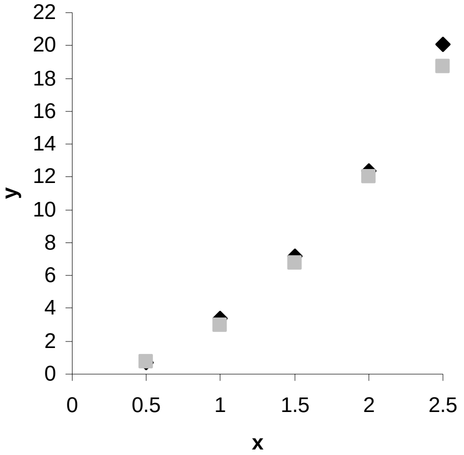


You may notice a key characteristic: each successive increase of the *y* coordinate is greater than the last. In other words, *y* grows more each time. There are several equations you can use to fit a model to data that graphs like this. The first one we'll look at is the proportion $y = kx^n$, where *k* is a constant and *n* is some exponent.

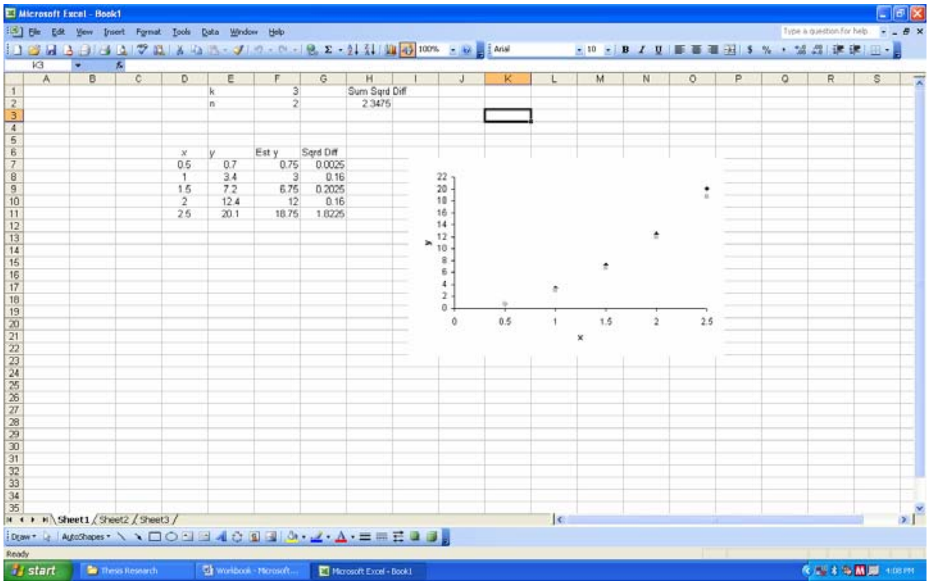
You face two challenges here. First, you need to determine what exponent to use for *n*. Generally, the higher the value of the exponent, the greater the growth. Since the growth of the data above is not too out of control, we might expect the exponent to be 2. This gives us the model $y = kx^2$. Next, you need to estimate what the constant will be. (We just want to get in the ballpark so Solver can do its thing.) The easiest way to estimate the constant is to solve for *k*. Plug your *x* and *y* data into your model and solve, like this

$$\begin{aligned} 0.7 &= k(0.5^2) \\ 0.7 &= k(0.25) \\ \frac{0.7}{0.25} &= k \\ 3 &\cong k \end{aligned}$$

We can estimate our constant to be 3. We can let Solver take it from there. So our model becomes $y = 3x^2$. What does that look like when compared to the data? Here's the graph



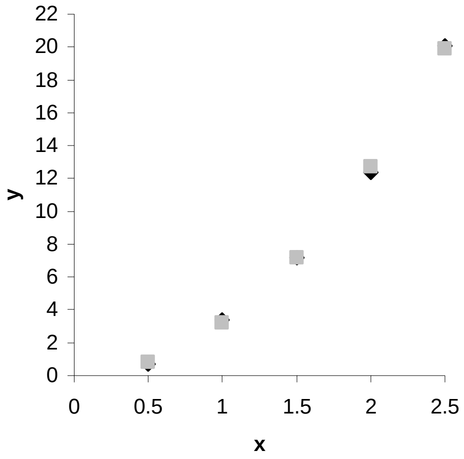
Not too shabby, but it's an estimate, so let's see what Solver can do for us. For ease of modeling, we don't want to allow solver to change our exponent (we don't want to have to raise something to the 2.11346th power), so let's leave it at 2. Everything else will be the same as the last example we did. First, we need an estimated *y* based on the model we used. Plug the model in and use the *x* values to get an estimated curve. Calculate the squared differences between the estimated curve and the actual one and then get the sum. Tell solver to minimize that sum by changing *k*. Here's what your spreadsheet should look like before you invoke Solver.



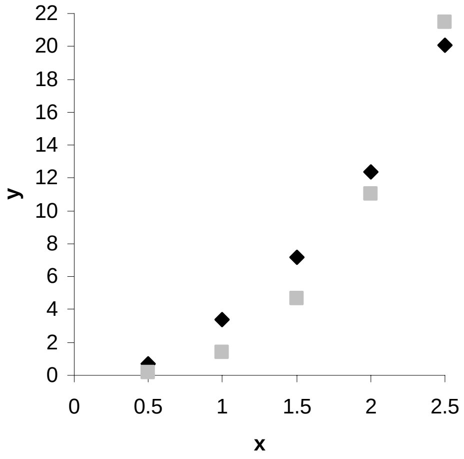
k	3		Sum Sqrd Diff
n	2		2.3475

x	y	Est y	Sqrd Diff
0.5	0.7	0.75	0.0025
1	3.4	3	0.16
1.5	7.2	6.75	0.2025
2	12.4	12	0.16
2.5	20.1	18.75	1.8225

After running Solver, we get a *k* of 3.19 and the plot looks like this

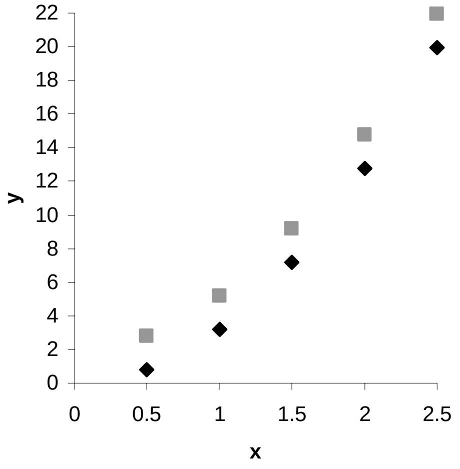


Not too bad. The model, $y = 3.19x^2$, is accurate and will allow precise predictions of any value of *x*. To ensure we used the right exponent, let's look at the best Solver could do if we had used x^3 instead of x^2.



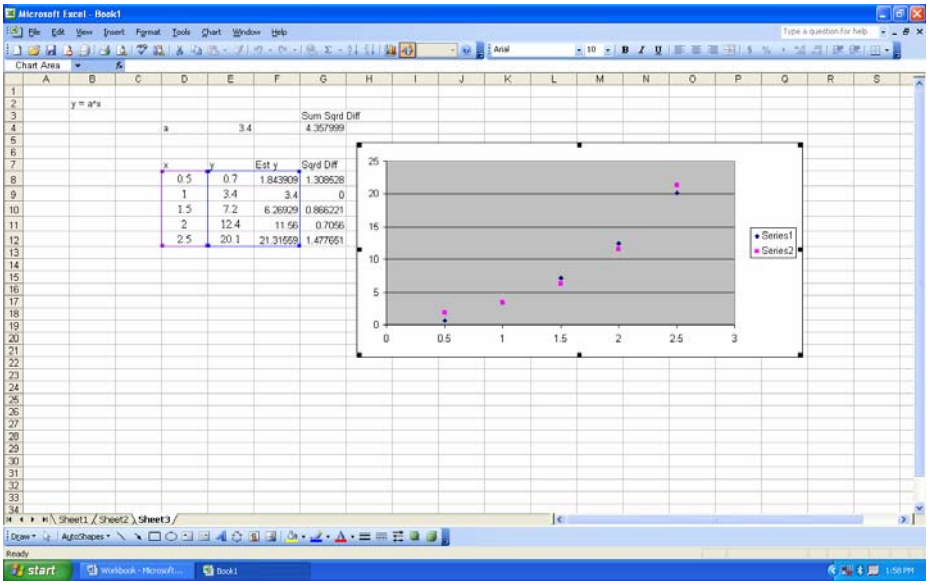
x^2 is obviously a better fit; we must have been on track. If however, after plotting the observed data, the growth rate appears higher, try cubing *x*.

What if the curve does not go through the origin? Take a minute to remember the model we used for a line when the *y* intercept was not zero. If you recall, we had to add the value of the *y* intercept to ensure the model accurately reflected the data. We'll do this again with a curve where the data does not go through the origin. The equation is $y = k_1x^n + k_2$. To show the difference between this and the previous equation, the graph below is based on the equation $y = 3.19x^2 + 2$. We've simply added 2 to the previous equation. Let's see what the graph shows



As we can plainly see, the plots are identical except that one is 2 units above the other. When *x* is 0, *y* will be 2. So, as with our line, when our curve doesn't cross the *y* axis at 0, we need to give a little boost. In this case the addition of + k_2 to our equation was the fix necessary. Of course, the value of k_2 will be whatever the *y* intercept is. Give this equation a try if the data seems to warrant it.

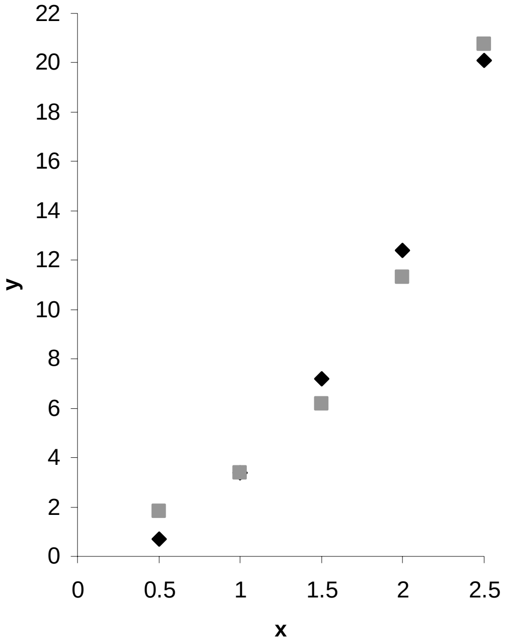
Another equation we can use to fit this type of curve is $y = k^x$. This equation is a bit different because we are choosing a constant, *k*, that, when raised to the power of our *x* coordinate, gives us the *y* coordinate. Once again we need to estimate the constant so we can have an estimated curve that Solver can work with. Luckily in our data, (we'll use the same data as before) we have the coordinates, (1, 3.4). We want a fairly close estimate so Solver will work for us. Using our equation, $y = k^x$, we can see that if we use the value 3.4 for our constant, *k*, and raise it to the power of *x*, in this case 1, the model will yield exactly the *y* coordinate or 3.4. Exact is a good place for Solver to start. Let's use that. Build a spreadsheet to let Solver find the "best fit" curve for this data. It should look like this.



y = a^x

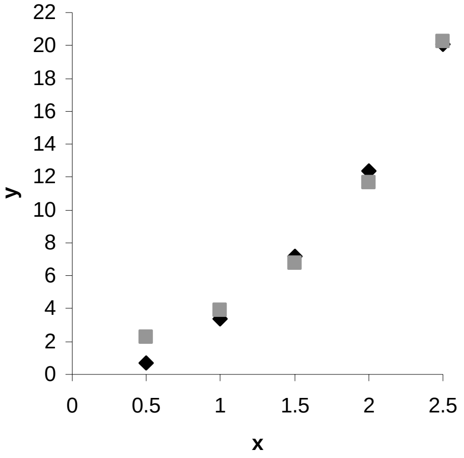
a		3.4	Sum Sqrd Diff 4.357999
x	y	Est y	Sqrd Diff
0.5	0.7	1.843909	1.308528
1	3.4	3.4	0
1.5	7.2	6.26929	0.866221
2	12.4	11.56	0.7056
2.5	20.1	21.31559	1.477651

Our curve is fairly close already. Let's let Solver do its thing to see how much better we can get. The new graph looks like this.

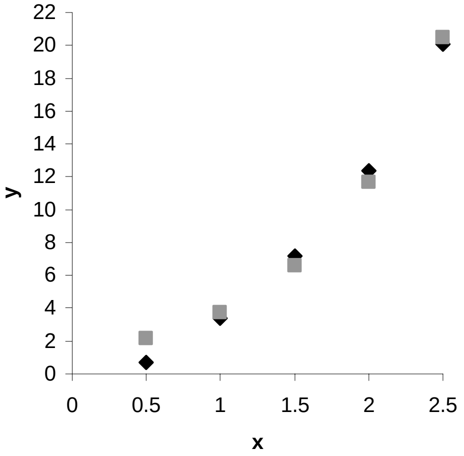


Solver gave us a value of 3.36 for *k* and a sum of squared differences of 4.95. Not too bad, but you can see that this model does not fit as well as $y = kx^n$. Another thing to note is the trend of the new curve at the beginning and ending of your data. In the graph above, the new line is clearly beginning to deviate quite a bit from the data. This would lead to quite a bit of error in prediction if we were to use this model, particularly out past either end of our actual data. You can see that if your data tends to curve up a bit more at the ends, this model would be a good choice.

Still another equation we can use to model exponential growth is $y = k_2 k_1{}^x$. This equation is similar to the last one we used, but we multiply k^x by another constant to tweak the curve just a bit. Again, to begin determining if this model is the best for our data, we need an estimated curve for Solver to massage for us. Using the data coordinates (2, 12.4) we can see that if we use 3 for the value of *k₁* (3^x or 3^2) a value of about 1.3 for *k₂* will get us close to 12.4 ($1.3 * 3^2 = 11.7$). This should give Solver a good place to start. Let's look at an estimated curve as compared to our data.



Not too bad. What does Solver give us?



Solver improves on our guess a little, giving a value for k_1 of 3.1 and a value for k_2 of 1.22, yielding a sum of squared differences of 4.26. This model seems to fit the data a little better overall, but take a look at that first point. Any prediction of *y* based on an *x* coordinate of less than about 1 will be way off. This model is probably not the best choice for this data. Use it in similar situations as the last one, but where you need a little tweak of the constant.

The best way to find a model that fits your observed data is to be familiar with the curves of several types of models. We've shown you a couple of types of curves along with models fitting data that graphs out to a line. As you work with these data and curves, you'll become more familiar with characteristics of each and picking the right model will get easier. Next lesson, we'll begin our look at simulation modeling.

THIS PAGE INTENTIONALLY LEFT BLANK

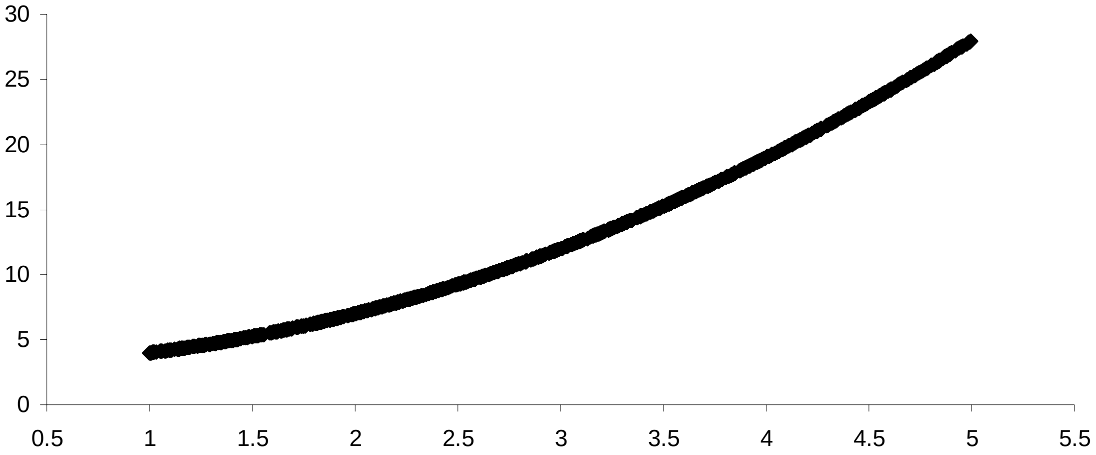
X. LESSON 9 – MONTE CARLO METHOD

There will be times when, for one reason or another, you won't be able to actually gather the data you need in order to build your model to make predictions. For example, if you're trying to make predictions about traffic patterns at roadblocks in a known insurgent zone, you wouldn't want to stop traffic just to gather data. This may needlessly endanger the lives of those involved with the roadblock. When you can't actually observe the behavior you want to model, you may need to simulate the behavior. The method of simulating behavior we'll look at is called Monte Carlo simulation.

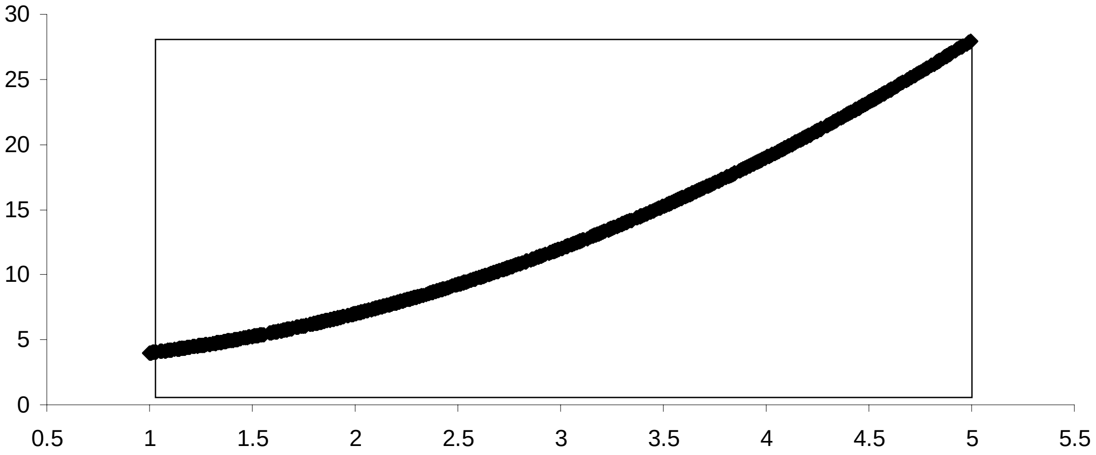
To start the lesson, we need to introduce two new terms: probabilistic and deterministic. Probabilistic refers to those processes in which there is some element of chance, like flipping a coin. You might think that when you flip a coin 100 times, it will come up 50 heads and 50 tails, but there is an element of chance so it's unlikely to come up 50/50. On the other hand, a process like finding the area of a geometric shape would be deterministic. There is no chance involved; the area is the area. Monte Carlo is a powerful probabilistic modeling tool. However, Monte Carlo simulation can be used to approximate a deterministic behavior.

We also need to introduce a new expression: $f(x)$. Pronounced "*f* of *x*" or "function of *x*", this expression is used when talking about graphed equations and generally refers to any equation involving the *x* coordinate; the <u>function</u> of *x*. Essentially, we're going to leave the "$y =$" off or our equations from now on. We'll say "the function of *x* is . . ." or "$f(x)$ is . . ."; think of $f(x)$ as a label for any equation involving the *x* coordinate.

Now, let's take a look at the Monte Carlo method. Monte Carlo uses random numbers to simulate occurrences of the behavior you are trying to model. These simulated occurrences can then be manipulated to best match the behavior you're studying in order to develop a model. To illustrate how Monte Carlo can be used, we'll look at a basic deterministic process – modeling the area under a curve. Suppose you have the function $f(x) = x^2 + 3$. This gives you a curve like this



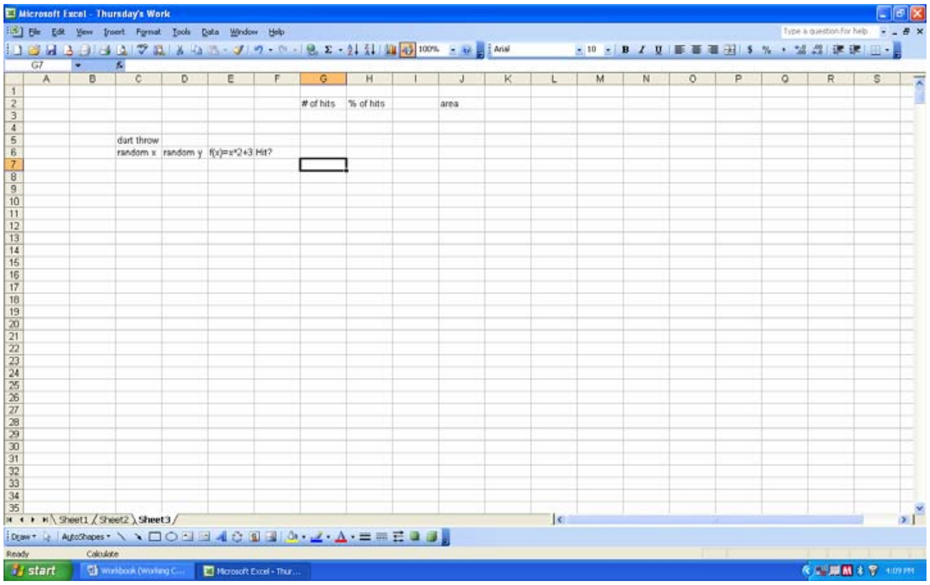
Also suppose the function satisfies $0 \le f(x) \le 28$ over the closed interval $1 \le x \le 5$. What this means is that the function $f(x) = x^2 + 3$ is bounded by a container 4 (along the *x* axis, from *x* = 1 to *x* = 5) by 28 (along the *y* axis, from *y* = 0 to *y* = 28). Like this



You are asked to find the area of the container under the curve. We intuitively know that the area under the curve is some part of the container. In fact, we can express the area just that way – as a percentage of the total area of the container. We can quickly determine that the area of the container is 112 (4 * 28), but what is the area under the curve? We'll use Monte Carlo simulation to find out.

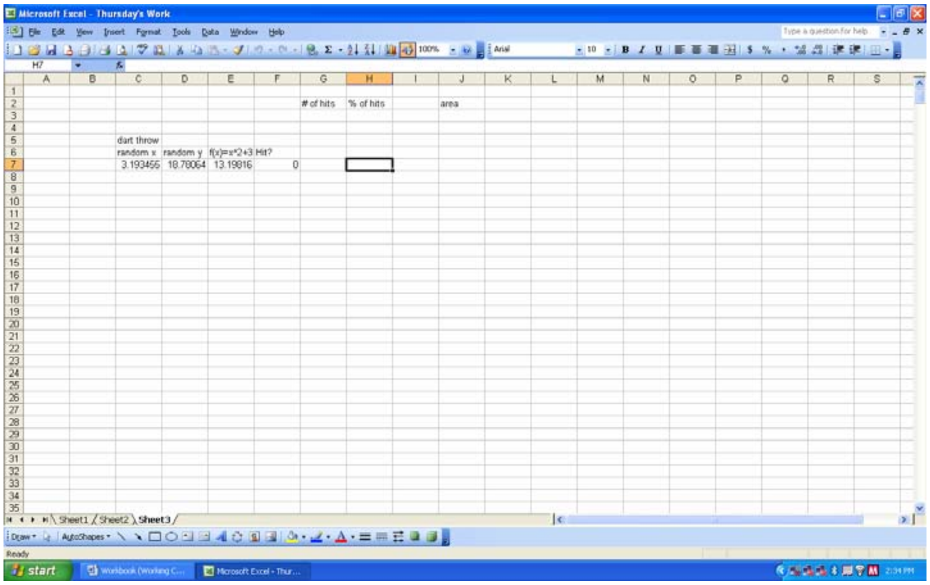
We're going to "throw darts" at our container. Actually, we're going to let Excel throw the darts for us by generating random numbers. All of the "darts" will land inside the container; some above the curve and some below. If we throw enough "darts", we will be able to count the total thrown and the total below the curve. We can then express the number of darts below the curve as a percentage of the total. We'll then use this percentage as the percentage of container below the curve. Let's "throw some darts."

First we need to set up our spreadsheet. We're going to need columns for our random x and y coordinates, our "dart throws". We'll also need a column for the y coordinated calculated using our function and our random x coordinate. This column, when graphed, will give us the curve for $f(x) = x^2 + 3$. Next we'll need a column that designates each dart throw as a hit or a miss. A hit means the random y coordinate is less than the calculated y coordinate for any given x coordinate. In other words the dart throw is below the curve. We'll tell Excel to return a 1 if random y is less than calculated y and a 0 if it is greater. We also need cells for the total number of hits, the percentage of hits (hits/total thrown), and finally the area under the curve (hit percentage * total area). After setting up these columns and cells, your spreadsheet should look like this



	C	D	E	F	G	H	I	J
2					# of hits	% of hits		area
5	dart throw							
6	random x	random y	f(x)=x^2+3	Hit?				

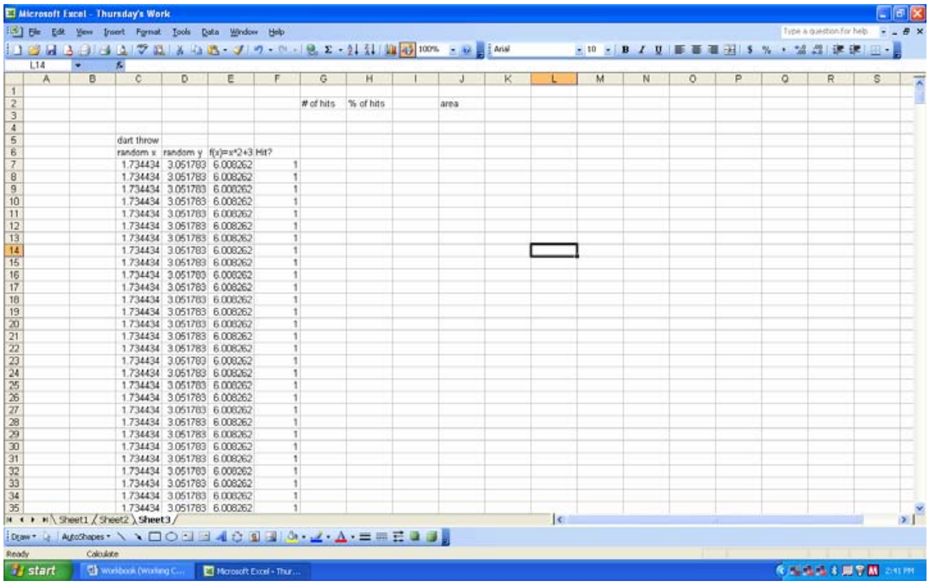
Now we need to generate random numbers to simulate our dart throws. This is Monte Carlo. We'll use the Excel function *Rand().* This function generates a random number between 0 and 1. We'll manipulate it slightly, because we need a different range of random numbers. For our random x we need a random number between 1 and 5. Because our random numbers have a range of 4, were going to multiply whatever Excel generates for us by 4. This gives us a random number between 0 and 4. Because our range starts at 1, we're going to add 1 to whatever Excel generates. This results in a random number between 1 and 5. The function will look like this *=rand()*4+1*. Type this in the first cell below your column heading *random x*. Now for the random *y*, we need a random number from 0 to 28. We simply need to multiply the random number Excel gives us by 28, like this *=rand()*28*. For the next column we need to calculate a value for *y* using the random *x* Excel generated for us. Write the formula to do the calculation in the appropriate cell. It should look like this *=C7^2+3*. For the *hit?* column we want Excel to return a value of 1 if the random *y* is less than the calculated *y* or 0 if it is greater. To do this, we will use an *If* statement. *If* statements work on the *if*, *then*, *else* principle. *If* a condition is true, *then* Excel returns a certain value, *else* it returns another value. The formula tells Excel what condition it's looking for, what to return if that condition is true and what to return if the condition is false. In our case, we want to evaluate the random *y* cell to see if it is less than the calculated *y* cell. If it is we want Excel to return a 1, if it isn't then we want Excel to return a 0. The formula follows the general form *=if(logical test, value if true, value if false)* and should look like this *=If(D7<E7,1,0)*. You're telling Excel: if the value in cell D7 is less than the value in cell E7 then display the number 1, if not (else) display the number 0. Of course, we'll write this formula in the cell where we want the 1 or 0 displayed. In our case it's the *hit?* column. Your spreadsheet should now look like this



Microsoft Excel - Thursday's Work
File Edit View Insert Format Tools Data Window Help
H7
Sheet1 Sheet2 Sheet3
Ready
start

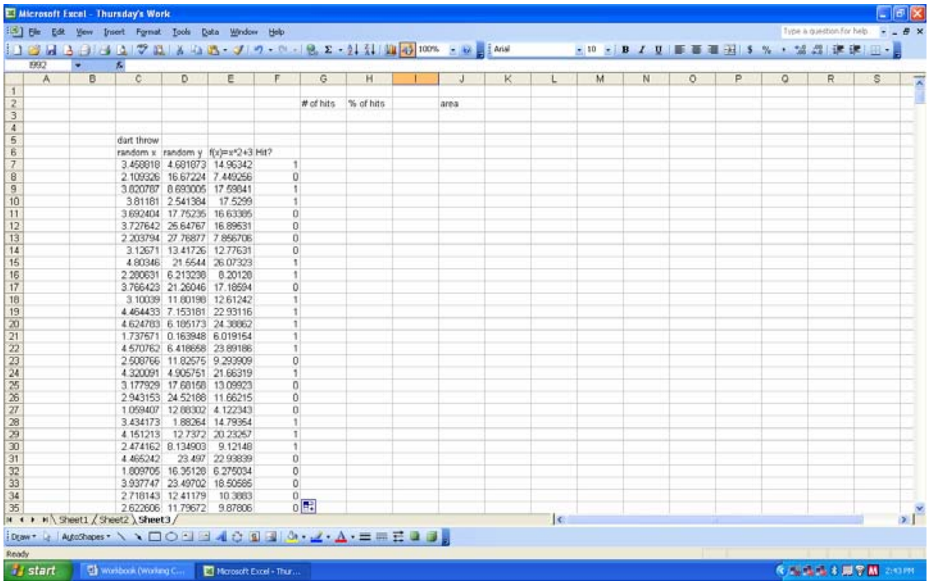
	A	B	C	D	E	F	G	H	I	J
2							# of hits	% of hits		area
5			dart throw							
6			random x	random y	f(x)=x^2+3	Hit?				
7			3.193455	18.78064	13.19816	0				

One note in passing here, the random number function (*rand()*) updates the cell values, i.e., generates a new random number, each time the spreadsheet is recalculated, like when you write a new formula for instance. You can manually generate new random numbers (and new values calculated from random numbers) by pressing the *F9* key. Try this now. You can see the random numbers being generated, check to ensure they appear to be within the ranges you specified, and check to ensure your *If* statement is working the way you want it to. Once you have all formulas right, copy the four columns down. We're gong to want quite a few "dart throws" to ensure an accurate estimation, so let's make about a thousand "throws". When you first do the copy action you'll notice all the cells are the same, like this


Microsoft Excel - Thursday's Work
File Edit View Insert Format Tools Data Window Help
L14
Sheet1 Sheet2 Sheet3
Draw AutoShapes
Ready
Calculate
start
Workbook (Working C...
Microsoft Excel - Thur...

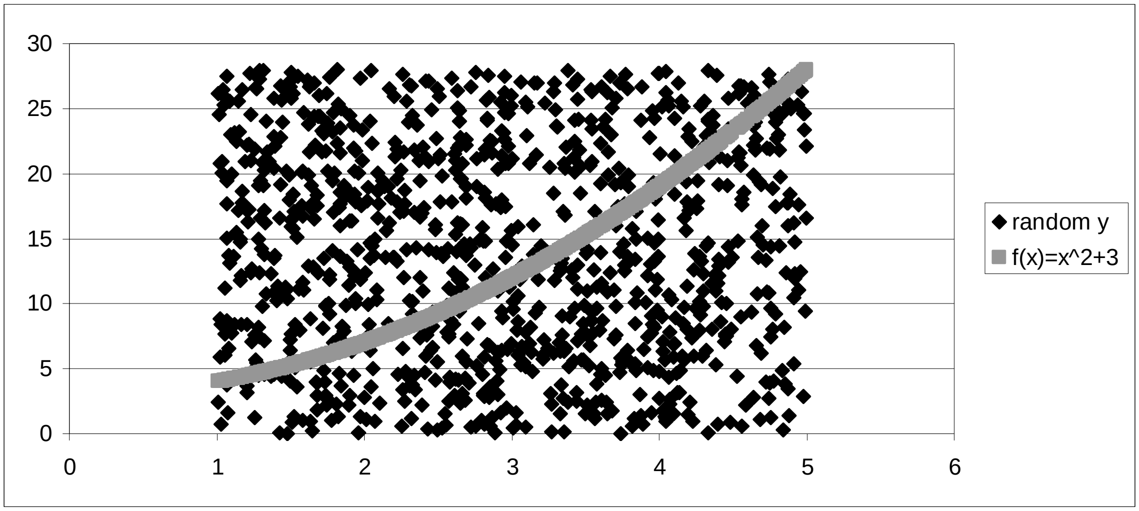
	A	B	C	D	E	F	G	H	I	J
1										
2							# of hits	% of hits		area
3										
4										
5			dart throw							
6			random x	random y	f(x)=x^2+3	Hit?				
7			1.734434	3.051783	6.008262	1				
8			1.734434	3.051783	6.008262	1				
9			1.734434	3.051783	6.008262	1				
10			1.734434	3.051783	6.008262	1				
11			1.734434	3.051783	6.008262	1				
12			1.734434	3.051783	6.008262	1				
13			1.734434	3.051783	6.008262	1				
14			1.734434	3.051783	6.008262	1				
15			1.734434	3.051783	6.008262	1				
16			1.734434	3.051783	6.008262	1				
17			1.734434	3.051783	6.008262	1				
18			1.734434	3.051783	6.008262	1				
19			1.734434	3.051783	6.008262	1				
20			1.734434	3.051783	6.008262	1				
21			1.734434	3.051783	6.008262	1				
22			1.734434	3.051783	6.008262	1				
23			1.734434	3.051783	6.008262	1				
24			1.734434	3.051783	6.008262	1				
25			1.734434	3.051783	6.008262	1				
26			1.734434	3.051783	6.008262	1				
27			1.734434	3.051783	6.008262	1				
28			1.734434	3.051783	6.008262	1				
29			1.734434	3.051783	6.008262	1				
30			1.734434	3.051783	6.008262	1				
31			1.734434	3.051783	6.008262	1				
32			1.734434	3.051783	6.008262	1				
33			1.734434	3.051783	6.008262	1				
34			1.734434	3.051783	6.008262	1				
35			1.734434	3.051783	6.008262	1				

Not to worry, Excel only copied, it hasn't recalculated. Press *F9* to recalculate and generate random numbers in your 1000 rows. Like this


Microsoft Excel - Thursday's Work
File Edit View Insert Format Tools Data Window Help
I992
Sheet1 Sheet2 Sheet3
Draw AutoShapes
Ready
start
Workbook (Working C...
Microsoft Excel - Thur...

	A	B	C	D	E	F	G	H	I	J
1										
2							# of hits	% of hits		area
3										
4										
5			dart throw							
6			random x	random y	f(x)=x^2+3	Hit?				
7			3.458818	4.681873	14.96342	1				
8			2.109326	16.67224	7.449256	0				
9			3.820787	8.693005	17.59841	1				
10			3.81181	2.541384	17.5299	1				
11			3.692404	17.75235	16.63385	0				
12			3.727642	25.64767	16.89531	0				
13			2.203794	27.76877	7.856706	0				
14			3.12671	13.41726	12.77631	0				
15			4.80346	21.5544	26.07323	1				
16			2.280631	6.213238	8.20128	1				
17			3.766423	21.26046	17.18594	0				
18			3.10039	11.80198	12.61242	1				
19			4.464433	7.153181	22.93116	1				
20			4.624783	6.185173	24.38862	1				
21			1.737671	0.163948	6.019154	1				
22			4.570762	6.418658	23.89186	1				
23			2.508766	11.82575	9.293909	0				
24			4.320091	4.905751	21.66319	1				
25			3.177929	17.68158	13.09923	0				
26			2.943153	24.52188	11.66215	0				
27			1.059407	12.88302	4.122343	0				
28			3.434173	1.88264	14.79354	1				
29			4.151213	12.7372	20.23257	1				
30			2.474162	8.134903	9.12148	1				
31			4.465242	23.497	22.93839	0				
32			1.809705	16.35128	6.275034	0				
33			3.937747	23.49702	18.50585	0				
34			2.718143	12.41179	10.3883	0				
35			2.622606	11.79672	9.87806	0				

Let's make a chart of our data so we can see what we've been doing. It should look like this



We've got our curve and all our dart throws. Now all we need to do is count the number of hits, divide by the number of throws and we'll have our percentage. Let's see how to make Excel do this for us.

We have the hits designated as 1's in our spreadsheet. If we simply total the *hit?* column, we'll have the number of hits. The formula for the sum of that column should look like this *=SUM(F7:F1006).* Press *F9* a few times. You'll see that the number of hits changes slightly each time, but not significantly.

Now we need the percentage of hits. This is simply the number of hits divided by the number of throws. The number of throws is, conveniently, the number of rows you copied. The formula should look like this *=$G$3/1000*.

Finally, we can calculate the area under the curve. We know the area of the entire container is 112 (4 * 28). We find the area under the curve by multiplying the area of the container by the percentage of hits below the curve. The formula should look like this =112*$H$3. Again, if you press *F9* a few times the value for the area will change slightly, but we can see that the area under the curve is about 54.

Generally, the more "throws" you make, the more accurate your estimation will be. For example, if we throw 5000 darts, the area varies much less and we can make a better estimate of our area, 53.5.

Monte Carlo simulation is an extremely versatile modeling tool. In this lesson we examined a classic use of Monte Carlo in modeling a deterministic process, the area under a curve. Next lesson we’ll look at using Monte Carlo simulation in modeling probabilistic processes.

THIS PAGE INTENTIONALLY LEFT BLANK

XI. LESSON 10 – SIMULATING PROBABILISTIC BEHAVIORS

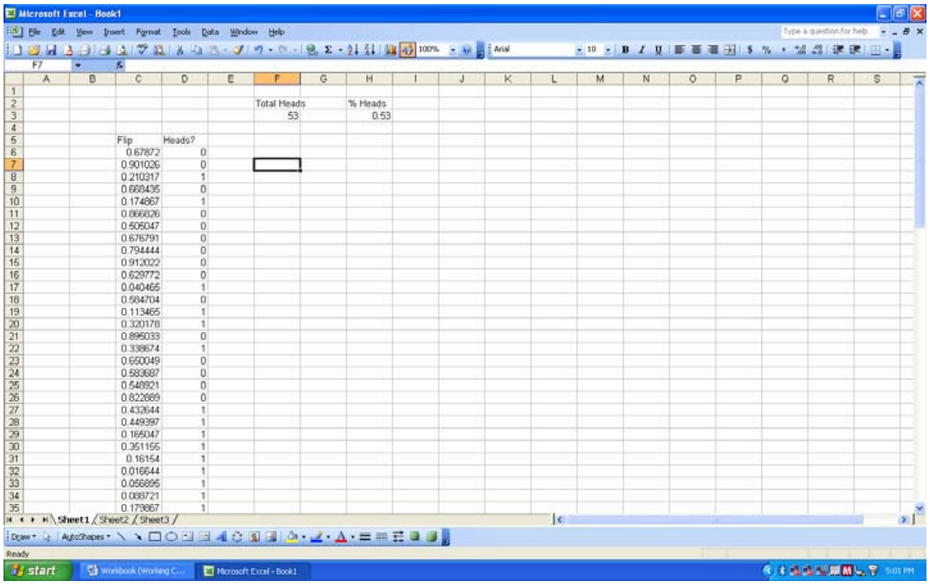
This lesson we're going to look into processes that are probabilistic, that is, processes that have some element of chance involved. Processes like flipping a coin or rolling a pair of dice are probabilistic. We'll also start to discuss probability. Probability refers to both randomness and uncertainty, but also deals with the likelihood of a certain outcome occurring.

Think of probability as a long-term average. If the probability of something happening is 1 out of 10, then in the long run the chance of that something happening is 1/10. Probability can be thought of as the ratio of

$$\frac{\text{number of particular events}}{\text{total number of events}}$$

We're going to model some simple probabilistic behaviors to further our understanding of probabilities. Let's get started.

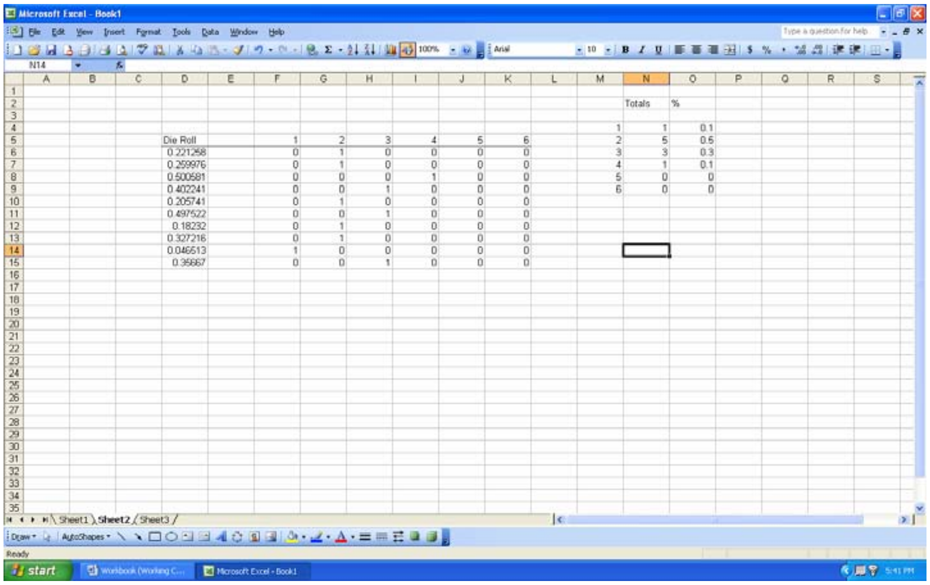
What are the chances of obtaining a head or tail when flipping a coin? Most of us will readily agree that the chance is 1 out of 2 or 1/2. Two sides to the coin, one desired outcome, easy. But, what really happens when we flip the coin? Let's let Excel flip a few thousand coins for us and find out. We know Excel can generate random numbers for us. The *Rand()* function will generate a random number between 0 and 1. Since we know that the chance of a head is 1 out of 2 or 1/2, we'll assign "Heads" to 1/2 of the random numbers *Rand()* generates. If *Rand()* generates a number less than or equal to 0.5 (1/2), we'll label it "Heads". In another column of our spreadsheet, we'll tell Excel to return a 1 if this is the case. If *Rand()* generates an number greater than 0.5, we'll label it "Tails" and have Excel return a 0. We can then sum the 1's to determine the total number of "Heads" and calculate the percentage of heads. For 100 flips, your spreadsheet should look something like this



	A	B	C	D	E	F	G	H
1								
2						Total Heads		% Heads
3						53		0.53
4								
5			Flip	Heads?				
6			0.67872	0				
7			0.901026	0				
8			0.210317	1				
9			0.668435	0				
10			0.174867	1				
11			0.866826	0				
12			0.505047	0				
13			0.676791	0				
14			0.794444	0				
15			0.912022	0				
16			0.629772	0				
17			0.040465	1				
18			0.584704	0				
19			0.113465	1				
20			0.320178	1				
21			0.895033	0				
22			0.338674	1				
23			0.650049	0				
24			0.583687	0				
25			0.548921	0				
26			0.822889	0				
27			0.432644	1				
28			0.449397	1				
29			0.165047	1				
30			0.351155	1				
31			0.16154	1				
32			0.016644	1				
33			0.056895	1				
34			0.088721	1				
35			0.179867	1				

To review, the formula for the random number generator is *=RAND()*, to return a 1 or a 0 the "If" statement is *=IF(C6<=0.5,1,0)*, to sum the "Heads?" column the formula is *=SUM(D6:D105)*, and the "% Heads" formula is *=F3/100*. Remember, when you press *F9* Excel will generate all new random numbers. It's just like flipping the coin 100 times again. Copy your formulas down to simulate 500 flips, 1000 flips, 10,000 flips. What percentages do you get for each set of flips? You should see the percentage get closer to 0.5 as the number of flips gets larger.

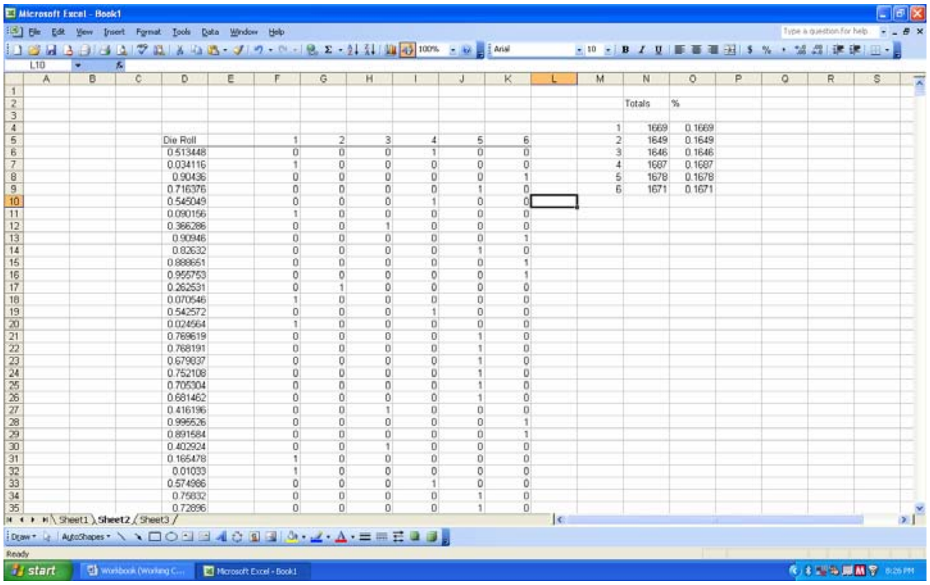
What if you have more than two possible outcomes? Rolling a fair die is a perfect example. In a fair die roll, each of the numbers has an equal probability of being rolled. With a six-sided die, that probability is 1/6. Again, we'll let Excel roll our die for us. We'll generate random numbers again, but this time we'll set up a system of counters for each value (1-6) using "If" statements, tally up each one and get percentages for each roll. Your spreadsheet should look something like this.



Die Roll	1	2	3	4	5	6
0.221258	0	1	0	0	0	0
0.259976	0	1	0	0	0	0
0.500581	0	0	0	1	0	0
0.402241	0	0	1	0	0	0
0.205741	0	1	0	0	0	0
0.497522	0	0	1	0	0	0
0.18232	0	1	0	0	0	0
0.327216	0	1	0	0	0	0
0.046513	1	0	0	0	0	0
0.35667	0	0	1	0	0	0

	Totals	%
1	1	0.1
2	5	0.5
3	3	0.3
4	1	0.1
5	0	0
6	0	0

The formulas for the counters are the only tricky piece of work here. There are a couple of ways to write them, but we'll use "If" statements with "And" functions. The "And" function let's you set two or more conditions and evaluate them as true or false. The general formula for the "And" function is *And(condition 1, condition 2)*. It will return true if all conditions are true and false if one or more is not true. The formulas for rolls of 1 and 6 are simple "If" statements: *=IF(D6<=1/6,1,0)* and *=IF(D6>5/6,1,0)*. The first formula says if the value of cell D6 is less than or equal to 1/6 then return 1, else return 0. The other one says if the value of cell D6 is greater than 5/6 then return 1, else return 0. The formulas for the other four counters are a bit more complicated. They look like this: for a roll of 2, *=IF(AND(D6>1/6,D6<=2/6),1,0)*; for 3, *=IF(AND(D6>2/6,D6<=3/6),1,0)*; for 4, *=IF(AND(D6>3/6,D6<=4/6),1,0)*; and for 5, *=IF(AND(D6>4/6,D6<=5/6),1,0)*. Once you have the counters set up, sum each column and calculate percentages for each roll. Notice in the spreadsheet above we rolled the die 10 times and the numbers 5 or 6 didn't show up once. This isn't exactly what we expect. We expect an even distribution. Let's have Excel roll our die 1000 times and see if we get closer to our expected percentages of 0.167. They should be a bit closer. Next, roll the die 10,000 times and check the percentages. Here's what the spreadsheet gives us.



Die Roll	1	2	3	4	5	6
0.513448	0	0	0	1	0	0
0.034116	1	0	0	0	0	0
0.90436	0	0	0	0	0	1
0.716376	0	0	0	0	1	0
0.545049	0	0	0	1	0	0
0.090156	1	0	0	0	0	0
0.366286	0	0	1	0	0	0
0.90946	0	0	0	0	0	1
0.82632	0	0	0	0	1	0
0.888651	0	0	0	0	0	1
0.955753	0	0	0	0	0	1
0.262531	0	1	0	0	0	0
0.070546	1	0	0	0	0	0
0.542572	0	0	0	1	0	0
0.024564	1	0	0	0	0	0
0.769619	0	0	0	0	1	0
0.768191	0	0	0	0	1	0
0.679837	0	0	0	0	1	0
0.752108	0	0	0	0	1	0
0.705304	0	0	0	0	1	0
0.681462	0	0	0	0	1	0
0.416196	0	0	1	0	0	0
0.995526	0	0	0	0	0	1
0.891584	0	0	0	0	0	1
0.402924	0	0	1	0	0	0
0.165478	1	0	0	0	0	0
0.01033	1	0	0	0	0	0
0.574986	0	0	0	1	0	0
0.75832	0	0	0	0	1	0
0.72896	0	0	0	0	1	0

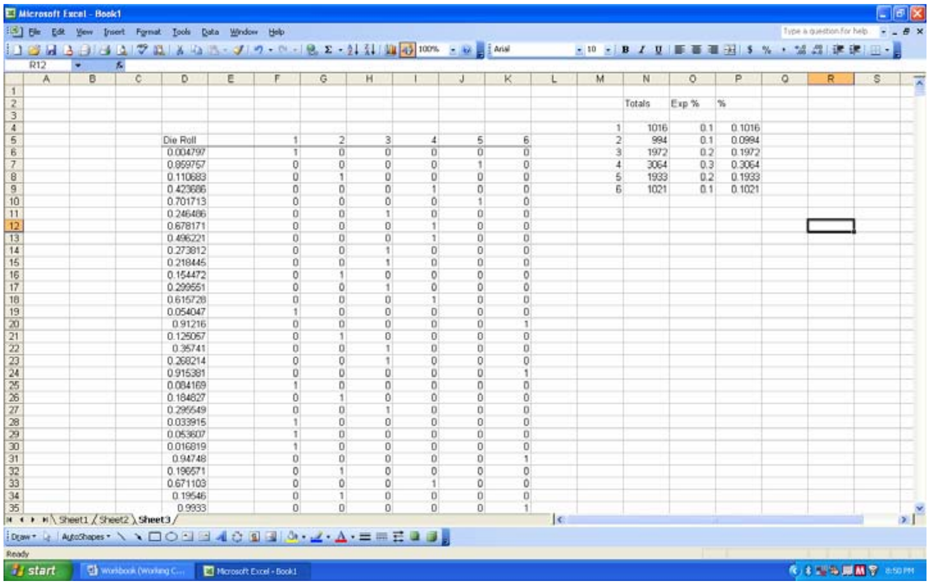
	Totals	%
1	1669	0.1669
2	1649	0.1649
3	1646	0.1646
4	1687	0.1687
5	1678	0.1678
6	1671	0.1671

Much closer to what we know the percentages should be. Let's look at one more example of a probabilistic process with more than two possible outcomes

In this example the likelihood of each event is not equal. A loaded die for example is more likely to come up on the "loaded" numbers than the other ones. Let's assume the weighted expectations are the following

Roll Value	1	2	3	4	5	6
Likelihood	0.1	0.1	0.2	0.3	0.2	0.1

In other words, the number 4 is 3 times more likely to come up than the number 1 and 1.5 times more likely to come up than the number 3. Let's let Excel roll this die for us. Remember, were using a random number between 0 and 1 to represent our die roll. We need to divide it up according to the weighted expectations in the table. Let's let 0 to 0.1 be 1, greater than 0.1 to 0.2 be 2, then greater than 0.2 to 0.4 will be 3. 3 is 2 times more likely than either 1 or 2 to come up, so we have to give it twice as much allocation in our random number. Number 4 will then be greater than 0.4 to 0.7 (3 times more likely). Greater than 0.7 to 0.9 will be number 5 and everything else (greater than 0.9) will be number 6. The set up for this spreadsheet will be very similar, use a column for your die roll, a set of counters for the numbers 1-6 using "If" statements and "And" functions where necessary, a sum for each number, and percentages for each number. Like this



Die Roll	1	2	3	4	5	6
0.004797	1	0	0	0	0	0
0.859757	0	0	0	0	1	0
0.110683	0	1	0	0	0	0
0.423686	0	0	0	1	0	0
0.701713	0	0	0	0	1	0
0.246486	0	0	1	0	0	0
0.678171	0	0	0	1	0	0
0.496221	0	0	0	1	0	0
0.273812	0	0	1	0	0	0
0.218445	0	0	1	0	0	0
0.154472	0	1	0	0	0	0
0.299551	0	0	1	0	0	0
0.615728	0	0	0	1	0	0
0.054047	1	0	0	0	0	0
0.91216	0	0	0	0	0	1
0.125067	0	1	0	0	0	0
0.35741	0	0	1	0	0	0
0.268214	0	0	1	0	0	0
0.915381	0	0	0	0	0	1
0.084169	1	0	0	0	0	0
0.184827	0	1	0	0	0	0
0.295549	0	0	1	0	0	0
0.033915	1	0	0	0	0	0
0.053607	1	0	0	0	0	0
0.016819	1	0	0	0	0	0
0.94748	0	0	0	0	0	1
0.196571	0	1	0	0	0	0
0.671103	0	0	0	1	0	0
0.19546	0	1	0	0	0	0
0.9933	0	0	0	0	0	1

	Totals	Exp %	%
1	1016	0.1	0.1016
2	994	0.1	0.0994
3	1972	0.2	0.1972
4	3064	0.3	0.3064
5	1933	0.2	0.1933
6	1021	0.1	0.1021

At 10,000 die rolls our percentages are very nearly the same as our expected percentages.

Alright, we've looked at three quick examples of how to use Monte Carlo simulation to model simple probabilistic processes. Use this simulation method anytime you need to model a system where there are chance outcomes and you cannot directly observe the system or conduct an experiment.

That's it! We're finished. By now, with the help of your instructor, you have mastered the basics of modeling. This quarter, we've given you the tools to analyze data and build a model for it. We've also laid the foundation for further study into the field of modeling. That will start next quarter. Good Luck.

In the next section you'll find a description of some pre-made models that will be provided to you on CD. This "toolkit" gives you some preformatted Excel spreadsheets to take with you so when you need a model you can look in the toolkit and very likely find what you need.

THIS PAGE INTENTIONALLY LEFT BLANK

XII. THE TOOLKIT

A. PURPOSE

Sometime after arriving at their next duty station Defense Analysis graduates realize that some of the modeling concepts they learned during their time at the Naval Postgraduate School (NPS) would be applicable to their current assignment. Unfortunately, for whatever reason, they had forgotten how to set up the appropriate models. These students would call back to the school for advice on how to setup the problem, or ask the department to help them with the problem due to time constraints. The intent of this toolkit is to provide examples and models in a generic format so that graduates can take it with them upon graduation to tackle just these sorts of problems. It is important to note that this is not intended to be a teaching module, but is intended for individuals who have already received instruction in modeling. The toolkit is intended to be a refresher or a template to solve problems at the next duty station.

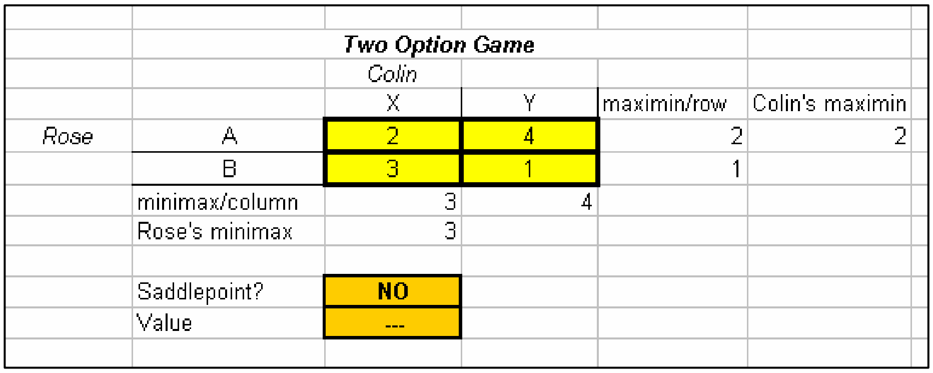
B. LIMITATIONS

It's important to note the limitations of modeling. First of all, they're just that, models. The fact that we have developed a model does not by definition mean that it is correct. Second, these models have been developed in a template format, and like most things with computers they're a function of operator input. In other words, the user would be wise to heed the old adage—"garbage in, garbage out." Finally, these are just computer files. They do not relieve you from actually thinking about the problem. For example, the graph below depicts a Lanchester Attrition model. The point of these equations is to determine the correct amount of forces required and the effectiveness of their equipment required to defeat the enemy. However, as you can see, the "Red Forces" continue to lose forces well into negative numbers. Another point to consider is that armies usually stop fighting well before they reach the last man. Further, it is extremely unlikely that the "Blue Force" would continue to increase their force structure when it is obvious that the "Red Force" is on the verge of collapse. My point is simply that you must continue to analyze the model, not take the output as the gospel truth.

C. USER INTERFACE

The models that follow cover a range of topics and model several different scenarios. Regardless of the intent, I have attempted to standardize the user interface as much as possible. Each folder may contain one or more excel files. Additionally, the files may have one or more tabs that serve different scenarios. I have attempted to make the naming convention as colloquial as possible. Inside the files the tabs are named at the top of the file with a brief introduction. Directions to the user have been highlighted in yellow.

Further, the cells that require user input are also highlighted in yellow with a bold border. The answer cells have been highlighted with an orange background. As much as possible, each file contains a graph so the user can view the behavior of the model.



	Two Option Game				
		Colin			
		X	Y	maximin/row	Colin's maximin
Rose	A	2	4	2	2
	B	3	1	1	
	minimax/column	3	4		
	Rose's minimax	3			
	Saddlepoint?	**NO**			
	Value	---			

XIII. THE MODELING FILES

A. DISCRETE DYNAMIC SYSTEMS

These are some of the most basic models. In them the dominant paradigm is future value = present value +/- change.

1. Drug Dosage_Rental Car.xls

a. Drug Dosage Problem

This model attempts to model the amount of a substance that decays at a certain rate. The problem is setup considering a drug administered to a patient. The drug needs to maintain a certain level of dosage in the system to remain effective. This model allows the user to vary the initial dosage, the decay rate, and the recurring dosage to maintain the correct dosage in the system. Further, the user can use Excel's "Solver" tool to determine the correct dosage.

b. Rental Car Problem

The rental car problem displays the relationship between two entities and the tendency for a variable to migrate to one entity or the other. The classic examples are the rental car scenario with two offices, two grocery stores in a small town, or the motor pool example from lesson 3 above. A percentage of the cars/patrons will stay loyal to one entity or the other, and the remainder will defect to the alternate entity. This model allows you to vary starting values and the return patronage percentage to determine how the model will behave.

2. Proportionality Problems.xls

This formula is rudimentary at its best. It is simply an example formula and how the user might determine the proportionality relationship between two objects.

3. Growth Models.xls

There are four tabs in this file.

a. Unconstrained Growth

The first, "Unconstrained Growth," shows how a population will increase or decrease based solely on their birth and death rates. The user can modify the starting population, birth rate, and death rate.

b. Constrained Growth

The second is a "Constrained Growth," model. This formula is determined by the difference between the growth and death rate of the population and then constrained by a max capacity or ceiling determined by the habitat. As you can see, the closer the population gets to the habitat "ceiling" the slower its growth rate becomes. The user can modify the starting population, growth rate, and habitat ceiling constraint.

c. Competing Series

The third is the "Competing Series" model. This model assumes that there are two species, i.e. owl and hawk, or two competing insurgencies, in a fixed area competing for resources. The user can modify the starting populations, their respective growth rates, and the death rate per specie interaction. This last figure is the most critical to the entire model. **NOTE:** it is important to keep this last figure very small because it is based on specie interaction. If the user enters a large number the model will produce errant data.

d. Predator-Prey Species

The final tab is the "Predator-Prey" model. This model looks at the relationship between two species that are not in competition with each other, but rather their population completely depends on each other to grow or decline. The example is the owl and mice. Without mice, the owl population with dwindle (due to lack of food) and conversely the mice will flourish in the absence of the owls. The user can input the starting values of species, the growth rate of the species, and the death rate/growth rate based on the two species interacting. Like the Competing species model, the interaction figures need to be very low or the model will produce errant data.

B. LANCHESTER MODELS

Lanchester combat models are concerned with the interaction between tow forces. These models were developed by F.W. Lanchester in an attempt to better understand air combat situations during World War I. These models allow the user to enter each forces' starting numbers and their combat efficiency or "kill rate." Each force, e.g., tank vs anti-tank weapons, loses forces based on the quantity of the opposing force. How each side's forces are decreased based on the force structure of the opposing unit. **NOTE:** It is important to understand that the models have been programmed to stop increasing or decreasing a force size after either force has reach a "0" force level.

1. Lanchester Models_6 tabs.xls

a. Simple Attrition

This is the most basic of the Lanchester models. Forces are decreased based solely on the "kill rate" and the amount of the opposing force.

b. Integer Model

This model is the basic Lanchester model, with the added function that only whole numbers can be counted. In other words, wounded soldiers are removed from the battlefield.

c. Simple with Reinforcements

This model is the same as the Simple Attrition model except each force receives a finite amount of reinforcements after each engagement.

d. Integer with Reinforcements

This model is also similar to the previous reinforcement model, except that there are not partial units, i.e. wounded soldier.

e. Simple with Dynamic Reinforcements

This is another simple model, except that the user can input a different level of reinforcements after different engagements. The user is not forced to enter a fixed number of reinforcements after each engagement.

f. Integer with Dynamic Reinforcements

This is the same as the Simple with Dynamic Reinforcements except that only whole units are considered.

g. Guerilla v. Conventional

This is the most complicated of the Lanchester models. The best way to think of this model is a conventional force opposing an insurgency. The two forces are decreased based on the other force; however, the conventional force can be decreased without an interaction with the enemy, i.e. roadside bomb, where the insurgents are only decreased when there is an interaction between the two forces.

C. EXPECTED VALUE

Expected value models attempt to predict the long-term outcome of a particular scenario or game. It is important to note that these formulas are not useful to predict a one-time outcome; they are representative of the long-haul scenario. For example, if a player was to sit at a roulette table for X amount of hours, he can expect a net value of X at the end of the night. The model will predict his average profit/loss; not the winning/loss of each individual spin of the wheel.

1. Job Interview Problem.xls

This model is the classic job interview problem. The scenario is setup that a college student, about to graduate, has three job interviews scheduled. There are four possible outcomes from each interview; no job, fair, good, and excellent job offers. Each job offer has an associated probability. Entering the monetary value of each job offer and their associated probability the model will work backwards and recommend what job offers the student should accept at each interview. Again, this is representative of several samples over a given period. Based on this, the model will sometimes predict a monetary value of a job interview that is not associated with a particular offer. This is due to the long-term nature of the model.

2. Insurance Costs_Bidding Costs_Roulette_Craps.xls

a. EV Samples

This tab has two examples in it; an Insurance scenario and Estimated Cost/Profit of Projects. These formulas serve as illustrative way to set up expected value problems. The insurance scenario takes the annual premium income and subtracts the product of the probability of a policy holder filing a claim times the average claim amount. This determines the insurance companies expected income from each policy. The second is a general contractor scenario. The contractor can bid on one of two projects. Each project has its expected profit, cost of submitting the bid, and the probability of winning the bid. Based on these figures he can choose which project to bid on.

b. US Roulette

This tab shows the expected values associated with a US roulette wheel. Based on the probability of each outcome, the model shows the user what the expected outcome is for a particular bet. The user can modify the amount of the bet and the payoff for each bet, black, red or green. This allows the user to determine the best strategy based on the payoffs.

c. Craps

This tab shows the probabilities for each outcome of a craps game. The user can modify the payoffs for each outcome to determine what payoffs are necessary for the game to become profitable.

D. DECISION THEORY

These models attempt to help the user make a better decision based on the problems criteria.

1. Linear Programming_Optimal Mix.xls

This file has three tabs that allow the user to determine the optimal mix of products, subject to constraints, that achieves the maximum payoff, or outcome.

a. 2 Products_2 Inputs

This file allows the user to determine the best mix of two products that each has two inputs. The user enters the input required for each product, the profit of each product, and the max available of each resource. The user then uses the solver function to determine the optimal output.

b. 4 Products_4 Inputs

This tab is the same as the previous except that it allows the user four products with up to four resources required.

c. 2 Products_3 Inputs Graph

This tab is for those users who need to see the math. This tab is the same as the previous two, except that it displays how to geometrically solve the problem. The slope of the profit line, depicted by the three orange lines, is moved along the axis until it achieves the maximum profit while still remaining below all three constraint lines.

E. TWO-PERSON GAMES

This section addresses the different scenarios possible in two-person games.

1. Pure_Mixed Strategies_Partial Conflict.xls

This file contains three tabs; Pure Strategies, Mixed Strategies, and Partial Conflict.

a. Pure Strategies

The pure strategies tab addresses a total conflict model. When one player wins the other loses: what is good for one player is not good for the other. The player must choose one option or the other; there is no middle ground of partial strategies. If both players end up reaching a common ground result this is called a saddle point. The user inputs the options and the model determines if there is a saddle point or not. The tab includes a 2 X 2, and a 3 X 3 game.

b. Mixed Strategies

This tab shows the same sort of two-person game, except that the two players, without a saddle point, are inclined to pursue a mixed strategy to achieve their optimal outcome. The classic model is a hitter vs. a pitcher. The user enters the payoffs in the matrix and the model will first determine if there is a saddle point or not. If there is, the mixed strategies portion of the formula will product skewed results. Assuming no saddle point exists, the model will determine the optimal strategies for each player and the predicted outcome.

c. Partial Conflict

Partial conflict problems results, unlike the previous problems, have different outcomes for each player. This tab depicts the classic “Prisoner’s dilemma” or “Arms Race” model. Both players could achieve a higher payoff by cooperating, but based on an absence of communication, each player is inclined to pure other strategies. When each player is expected to reach the same conclusion this is called the Nash equilibrium. Based on the user inputs, this model will show how each player will be inclined to act, and determine if there is one or more Nash equilibriums.

d. Threat Promise

This model takes the partial conflict through the threat promise relationship. This model will first tell you if there is any Nash equilibrium’s like the previous model. However, in addition to any equilibrium, this model will also advise the user if either player has a strategic move. That is to say, if a player makes a move prior to the opponent can he/she achieve an outcome that is favorable to any equilibrium? Further, it examines the game to determine if either player has a promise, whether that promise will be seen as credible, whether either player has a threat, and whether that threat will be seen as credible or not.

e. Nash Arbitration

This model simplifies the task of determining the Nash Arbitration point. Unfortunately, this model will not determine the status quo point—there are just too many different ways to determine this point. As such, the user will need to enter the status quo point, and the North-West and South-East of the Pareto Optimal line. Once these values are entered the model will quickly and automatically determine the Nash Arbitration point and present a graphical display of the point.

F. MODELING FORMULAS

The intent of these four files is to assist the user in determining which modeling formula meets a system they are trying to model. The user should first enter the data they have collected and into the appropriate cells. Running the solver tool, which is preformatted for the formula, the user can see if this particular formula is well suited to modeling the system. If the graphs do not coincide the user can try the next formula. **NOTE:** it is important to note that solver works best when the numbers in the variable cells are somewhat close to the solution. The user may have to modify the inputs to get solver, "in the ballpark."

G. BONUS MATERIALS

This folder contains three separate files. The first contains an annuity and investment file. The second has examples of installment loans and mortgages. The third is a negotiation scenario taken from the "Strategic Thinking" book.

1. Annuity and Investment.xls

a. Annuity

The annuity tab allows the user to observe the performance of an annuity fund over time. This model allows the user to modify the starting amount of the principal, the interest rate, and the monthly withdrawal. The user can also use the solver function to determine how much the investor could withdraw every month in order to maintain the fund, or in order to deplete the fund after X periods. Likewise, the principal and interest rate can be modified to determine required parameters for the investor.

b. Investment

The investment tab allows the user to observe the performance of a simple compound interest account. The user can modify the starting amount, monthly deposit and the interest rate.

2. Installment Loan and Mortgage.xls

a. 60-month Installment

This model allows the user to modify all the factors affecting an installment loan for 60 months; purchase price, down payment, interest rate, and payment. Excel solver can also be used to determine the payment required for a particular purchase. Additionally, the file shows the user how much of each payment goes towards interest and how much towards principal.

b. 48-month Installment

This is the same as the previous file except it is setup for 48 months.

c. 72-month Installment

This is the same as the previous file except it is setup for 72 months.

d. Mortgage 30-year

As the name suggests, this tab analyzes a 30-year mortgage. It works the same as the previous tabs. Use the solver tool to determine what your monthly mortgage payment will be.

3. Negotiations.xls

This scenario is taken from the "Thinking Strategically: The competitive Edge in Business, Politics, and Everyday Life." book by Dixit and Nalebuff. In this scenario the Union and Management are negotiating over profit sharing for the union. The model is very simplistic in that there are 115 days available to work/make a profit and each side is only allowed one offer or counteroffer per day. The model shows that whoever is able to make the last offer is in the more powerful position. Also looking forward and reasoning back allows each party to see what "settlement" they ought to be satisfied with on each day.

a. Straight

This tab shows the negotiations where neither party has any alternative to their current situation. The union has no alternate source of income and management cannot make a profit without union employees.

b. Alternate Income Union

This tab shows how the negotiations change when the union has an alternate source of income. The fact that they can make some money even while they strike and management continues to loss all available profits puts the union in the power position. The user can change the amount of the union's income to see to what degree this changes the negotiations.

c. Alternate Income Both

This tab looks at the negotiations when both parties have an alternate source of income. The union can find part-time employment elsewhere while they hold out for profit sharing and the management can hire less efficient scabs at a reduced profit. In this scenario the formula shows just how important the alternate source of income is to negotiating from a position of power.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California